A Scalable Password-based Group Key Exchange Protocol in the Standard Model

David Pointcheval

École normale supérieure
& CNRS

Joint work with:
Michel Abdalla

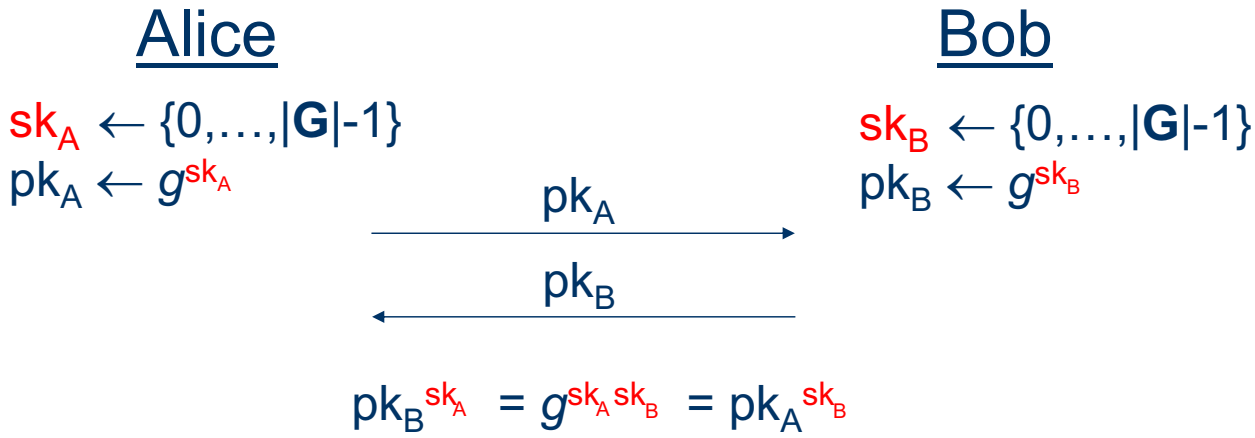
Authenticated Key Exchange (AKE)

Goal: Secure channel

- Allows two parties to establish a common secret in an authenticated way
- Intuitive goal: **implicit authentication**
  - The session key should only be known to the parties involved in the protocol
- Formally: **semantic security**
  - the session key should be *indistinguishable* from a random string

Diffie-Hellman Protocol

Let **G** be a group in which the DDH problem is hard and let g be a generator for **G**



$$pk_B^{sk_A} = g^{sk_A sk_B} = pk_A^{sk_B}$$

Protocol does NOT provide authentication

Authentication Techniques

- **Asymmetric techniques**
  - Assume the existence of a public-key infrastructure
  - Each party holds a pair of secret and public keys
- **Symmetric techniques**
  - Users share a random secret key
  - **2-party** or **3-party** settings
- **Password-based techniques**
  - Consider the case of weak secrets (e.g., a 4-digit PIN)

Group Password-based AKE (GPAKE)

- **Scenario**

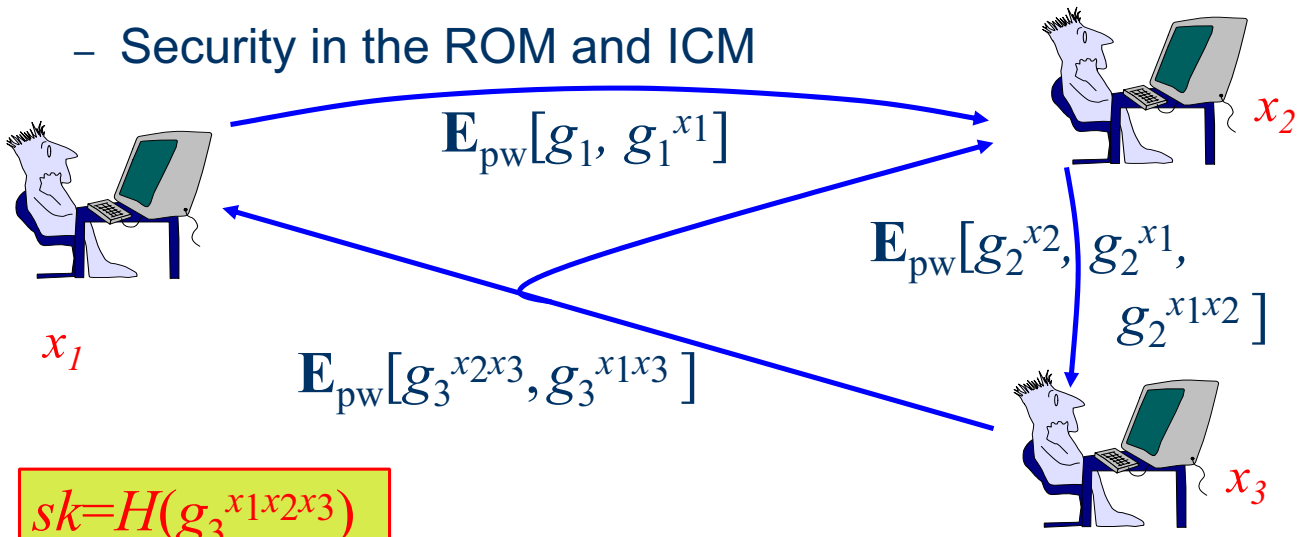
  Similar to the 2-party case, except that …
  - Number of protocol participants is variable
  - Password is shared among all participants
  - Session key is shared among all participants
- **Security goal**
  - **Similar to the 2-party case**: Indistinguishability
    Allows a pool of users to established a common session key with only the help of passwords

Communication Model

- Users can have many protocol instances running concurrently
- Communication controlled by the adversary
  - Adversary can create, modify, or forward messages
  - The transmission of messages is done via specific oracle queries

Previous Work on GPAKE

- **[BressonChevassutP02]**:
  - ***Group Diffie-Hellman*** password-based key exchange
  - Linear number of rounds
  - Security in the ROM and ICM

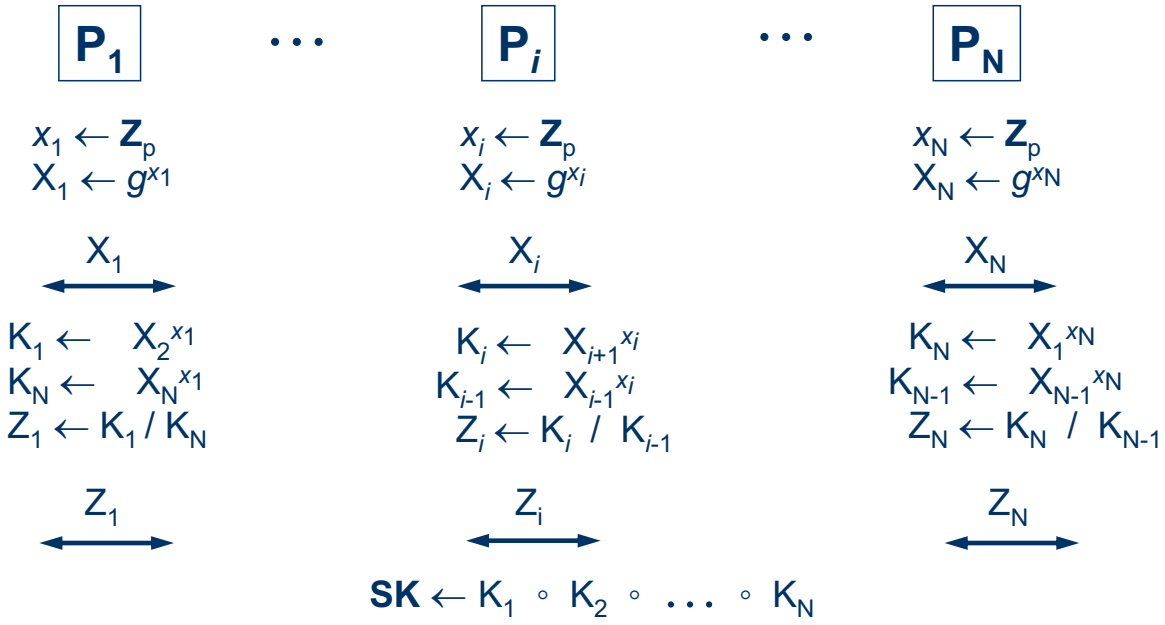


Previous Work on GPAKE

- **[LeeHwangLee04], [DuttaBarua06]**
  - Both based on the ***Burmester-Desmedt protocol***
  - *Both proven secure* in the ROM and ICM
  - **Both broken in [ABCP06]**
- **[AbdallaBressonChevassutP06],[TangChoo06]**
  - Based on the ***Burmester-Desmedt protocol***
  - Proven secure in the ROM and ICM

  Constant-round

The Burmester-Desmedt GKE (BD94)



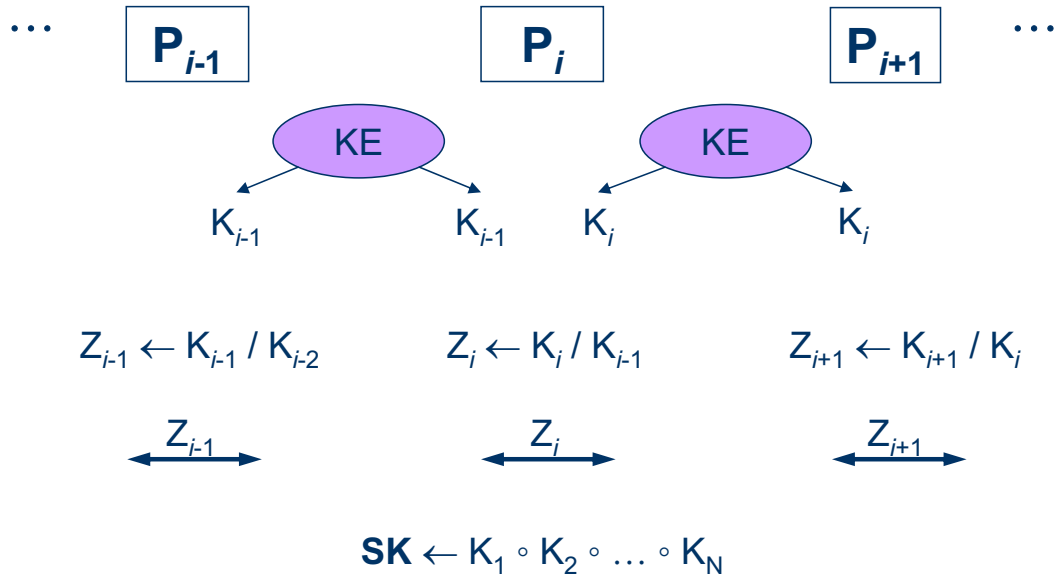
Protocol does NOT provide authentication

Adding Password Authentication Ideal Cipher Model

- **EKE approach**
  - Encrypt all flows using the password **pw**
  - In both and $\mathbf{X}_i = \mathrm{E}_{\mathbf{pw}}(X_i)$ and $\mathbf{Z}_i = \mathrm{E}_{\mathbf{pw}}(Z_i)$
- **Problem**
  - In the BD protocol, $Z_1 \circ Z_2 \circ \ldots \circ Z_N = 1$
  - Dictionary attack: Guess password **pw**
    - Compute $Z_i = \mathrm{D}_{\mathbf{pw}}(\mathbf{Z}_i)$ for i=1,…,N
    - Check if $Z_1 \circ Z_2 \circ \ldots \circ Z_N = 1$
- A **provably secure** approach: **[AbdallaBressonChevassutP06]**
  - Encrypt only the first round of the BD protocol
    With a key that depends on the password
    but also the session ID and the party ID

The Burmester-Desmedt GKE
A Generic Version

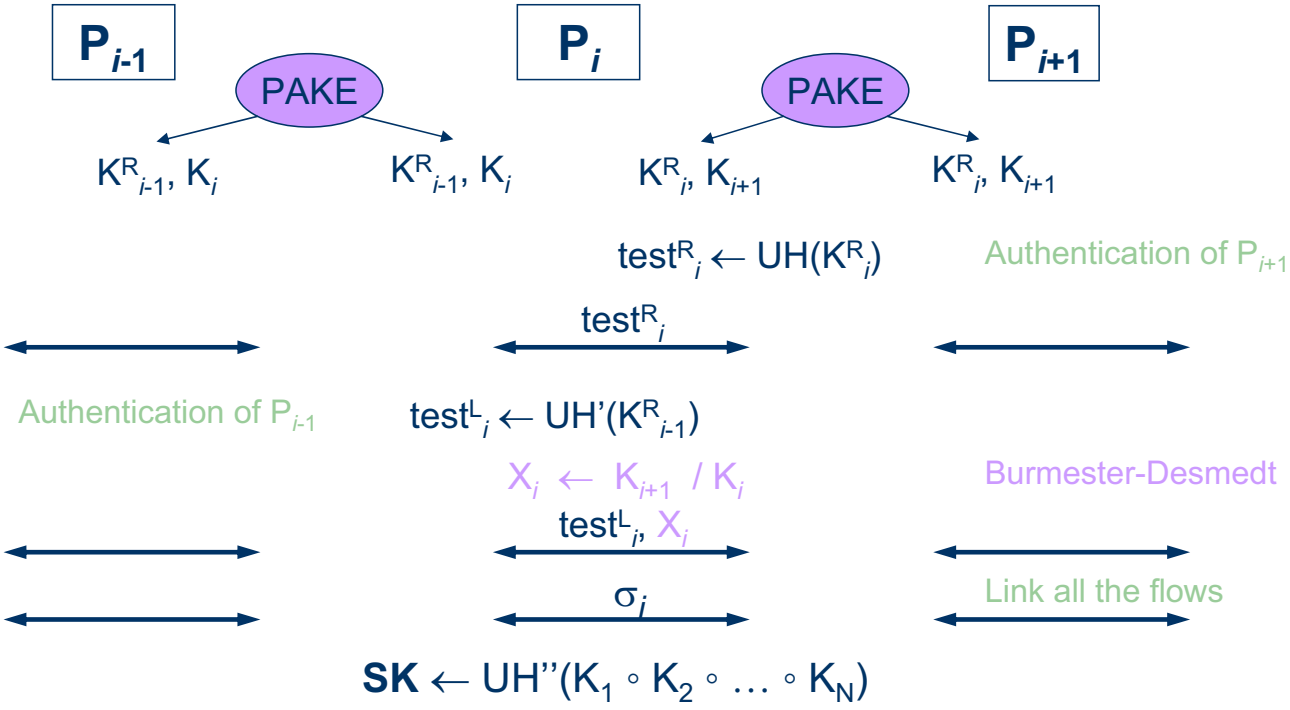
From any key exchange protocol KE:



A GPAKE in the Standard Model
Intuition

- Run an instance of the PAKE protocol between any two consecutive users
  - so that it generates 2 pairwise keys
- Each user should authenticate its predecessor and successor (using one of the pairwise keys)
- Use the 2 other pairwise keys to generate group session key (Burmester-Desmedt)
- Signatures authenticate the transcript of all messages that were broadcast in previous rounds, and that have to be linked together

A GPAKE in the Standard Model
Outline



Smooth Projective Hash Functions
[Gennaro-Lindell's variant]

- **Hash key generation:** hk = HK(pk)
  - pk – public encryption key, hk – hashing key
- **Projected key generation:** hp = α(hk, c)
  - hk – hashing key, hp – projected key, c = E(pk,m;r) – ciphertext
- **Hashing algorithm:** H (hk, m, c) $\in$ G
  - m – message, c = E(pk,m;r) – ciphertext, hk – hashing key
- **Projected hashing algorithm:** h = h(hp, m, c; r)
  - hp – projected key, r – random coins, c = E(pk,m;r)

Smooth Projective Hash Functions Security Properties

- **Correctness:**
  - If c = E(pk,m;r), then
    (m,c,hp = α(hk, c)) uniquely determines H(hk,m,c)
  - When c = E(pk,m;r), then
    H(hk,m,c) can be computed efficiently given r

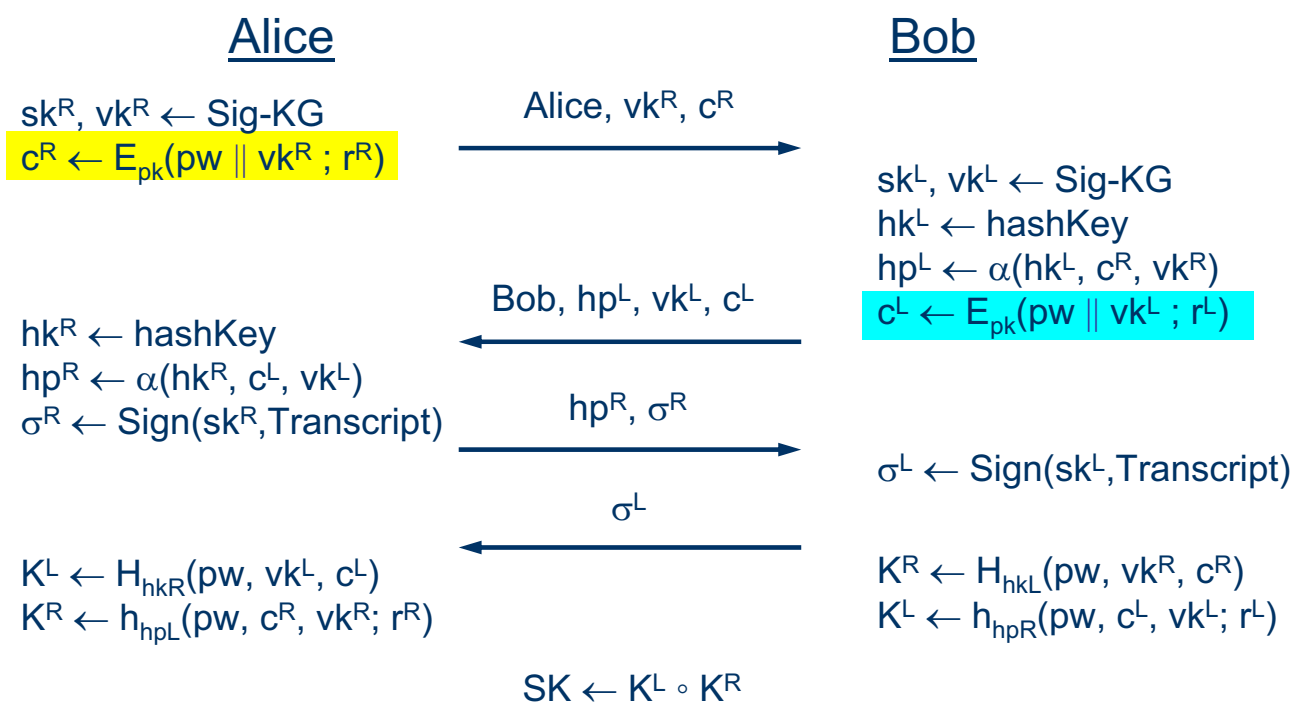
    h(hp,m,c; r) = H(hk,m,c)
- **Smoothness:** (*statistically*)
  - If c is not an encryption of m, then
    (m, c, hp) gives **no** information on H(hk,m,c)
- **Pseudo-randomness**: (*computationally*)
  - When c=E(pk,m;r) and hp=α(hk,c), then
    H(hk,m,c) is pseudo-random given (m,c,hp)

The Gennaro-Lindell Construction

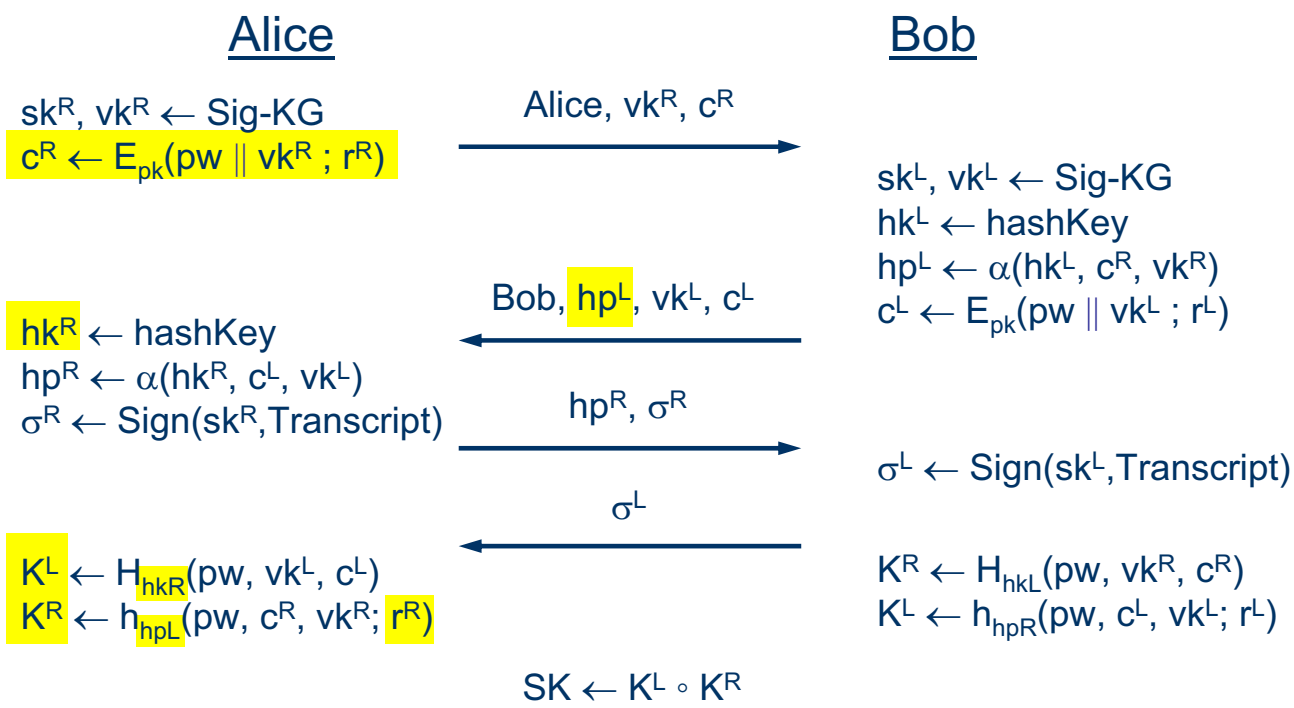
Alice		Bob
$sk^R, vk^R \leftarrow$ Sig-KG $c^R \leftarrow E_{pk}(pw \,\|\| \, vk^R ; r^R)$	$\xrightarrow{\text{Alice}, vk^R, c^R}$	
		$sk^L, vk^L \leftarrow$ Sig-KG $hk^L \leftarrow$ hashKey $hp^L \leftarrow \alpha(hk^L, c^R, vk^R)$ $c^L \leftarrow E_{pk}(pw \,\|\| \, vk^L ; r^L)$
$hk^R \leftarrow$ hashKey $hp^R \leftarrow \alpha(hk^R, c^L, vk^L)$ $\sigma^R \leftarrow$ Sign(sk^R,Transcript)	$\xleftarrow{\text{Bob}, hp^L, vk^L, c^L}$	
	$\xrightarrow{hp^R, \sigma^R}$	$\sigma^L \leftarrow$ Sign(sk^L,Transcript)
	$\xleftarrow{\sigma^L}$	
$K^L \leftarrow H_{hkR}(pw, vk^L, c^L)$ $K^R \leftarrow h_{hpL}(pw, c^R, vk^R; r^R)$		$K^R \leftarrow H_{hkL}(pw, vk^R, c^R)$ $K^L \leftarrow h_{hpR}(pw, c^L, vk^L; r^L)$

$SK \leftarrow K^L \circ K^R$

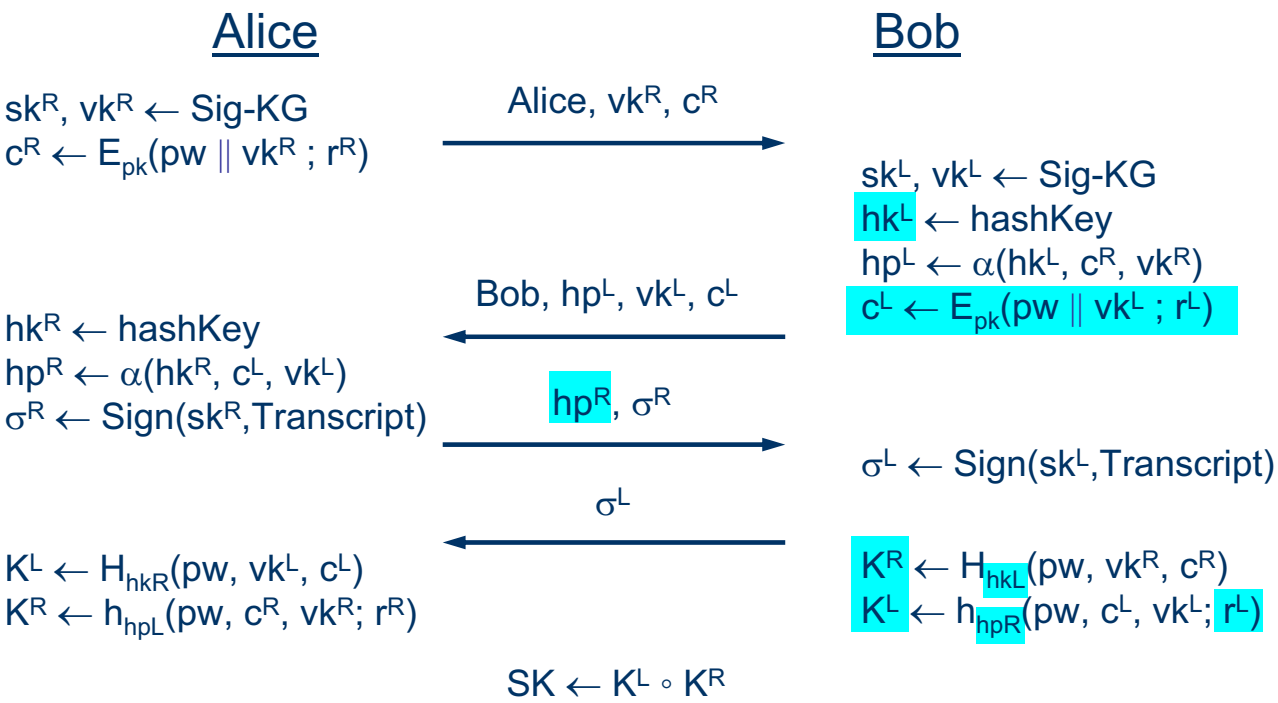
The Gennaro-Lindell Construction



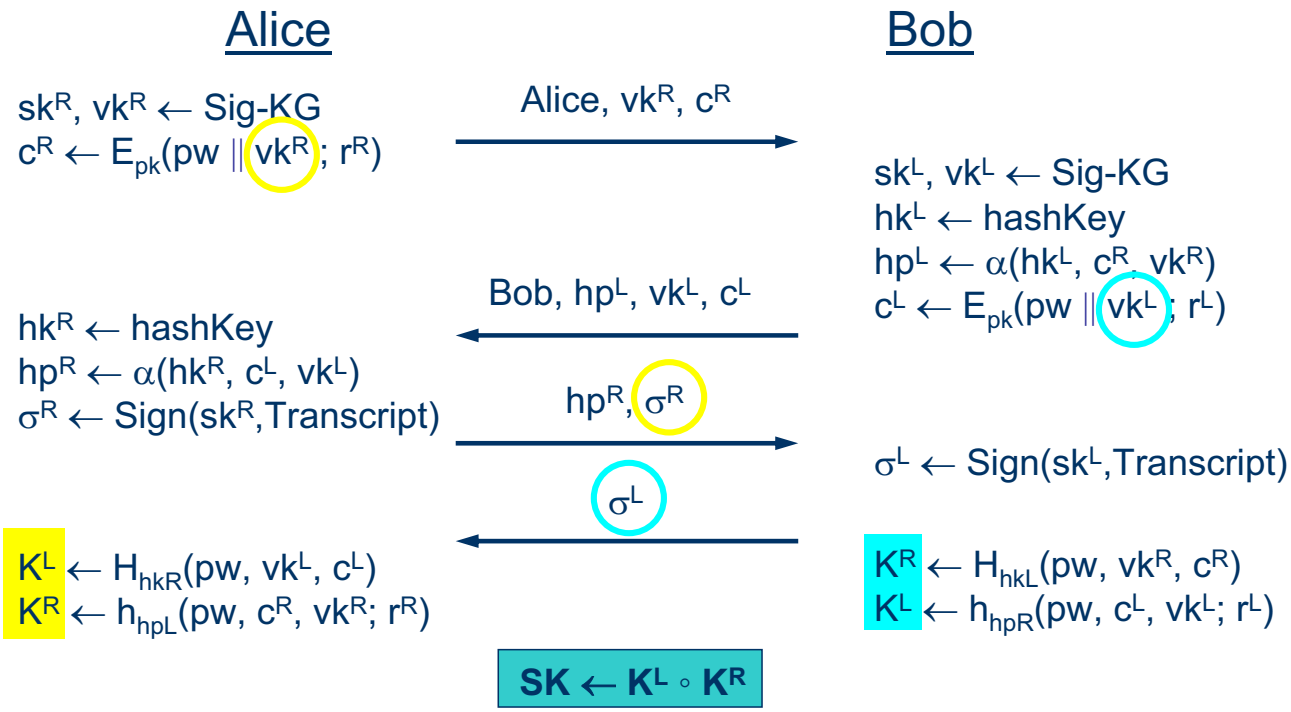
The Gennaro-Lindell Construction



The Gennaro-Lindell Construction

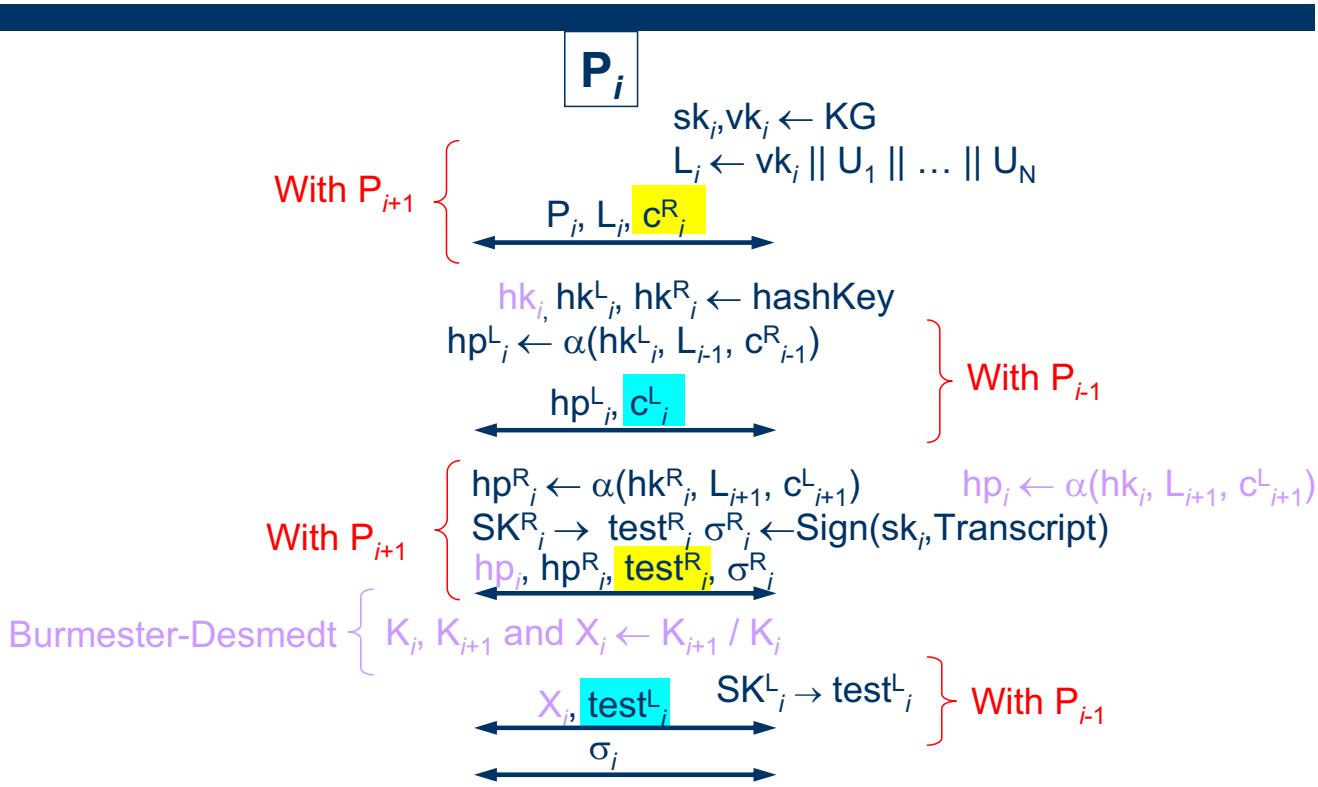


The Gennaro-Lindell Construction



A Scalable Password-based Group Key Exchange Protocol in the Standard Model

A GPAKE in the Standard Model Details



A GPAKE in the Standard Model Security

- IF
  - LPKE is a labeled encryption IND-CCA
  - HASH is a family of smooth projective hash functions
  - UH, UH', UH'' are families of universal hash functions
  - SIG is a signature scheme SUF-CMA (2-time secure)
- THEN
  - The protocol described in the previous slides is a secure GPAKE protocol

$$Adv \leq O(q_{send} / D) \leq O(N\, q_{session} / D)$$

Concluding Remarks

- Efficient GPAKE
  - 5 rounds
  - 2 encryptions, 3 projections
  - 3 hashings, 3 projected hashings
  - 5 universal hashings
  - 2 signatures, N verifications: 2-time signatures
- Secure GPAKE in the standard model
  - Under classical assumptions (DDH, QR, HR)
- TCC07: **[AbdallaBohliGonzalezSteinwandt07]**
  - Generic compiler from 2-party to group AKE
  - With the same authentication mode
  - Proven secure in the standard model