ETH zürich

Forward-secure 0-RTT Key Exchange from Puncturable Key Wrapping

Felix Günther

based on joint work with Matilda Backendal and Kenny Paterson, in submission

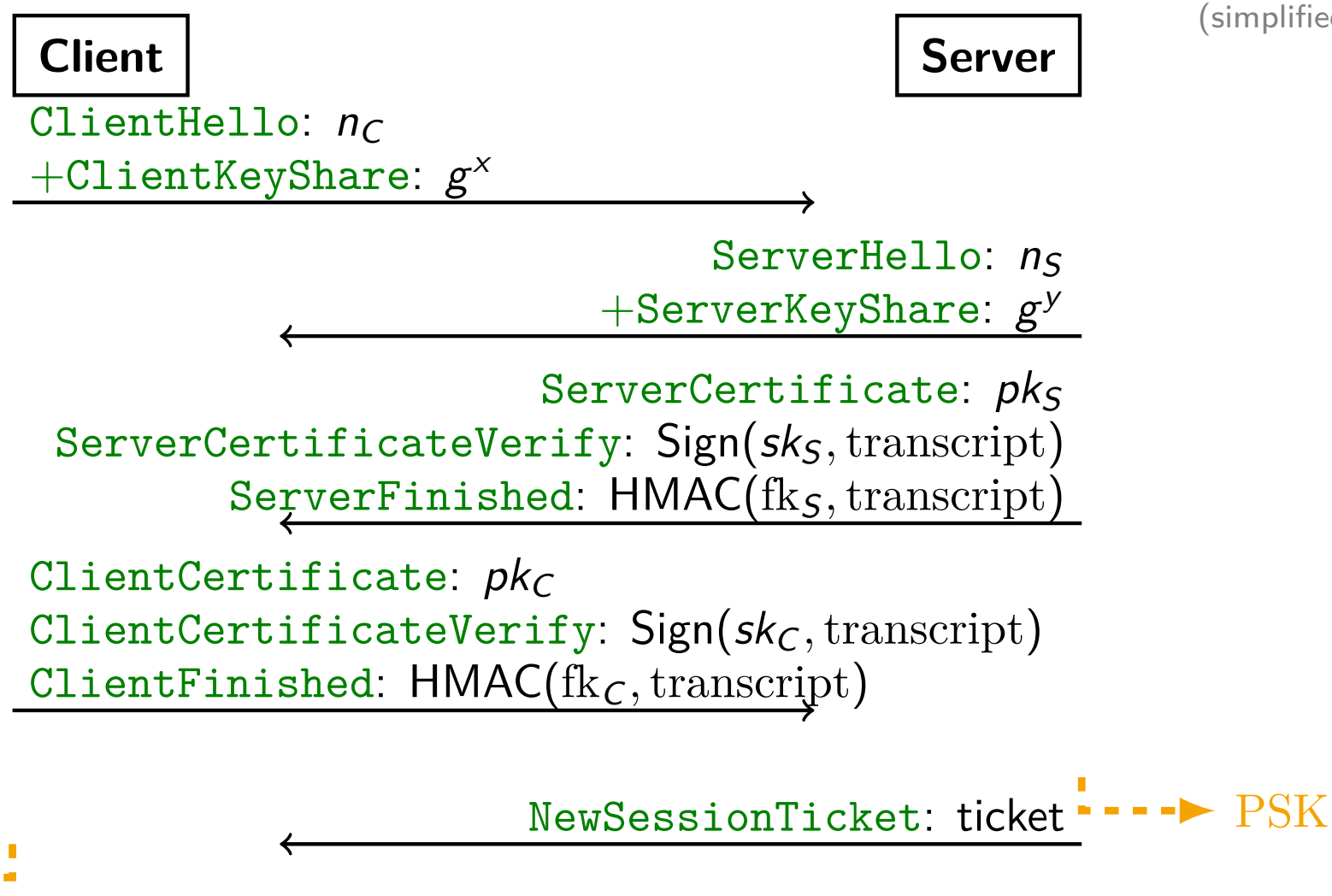
ETH zürich

APPLIED CRYPTO GROUP

The TLS 1.3 Full Handshake

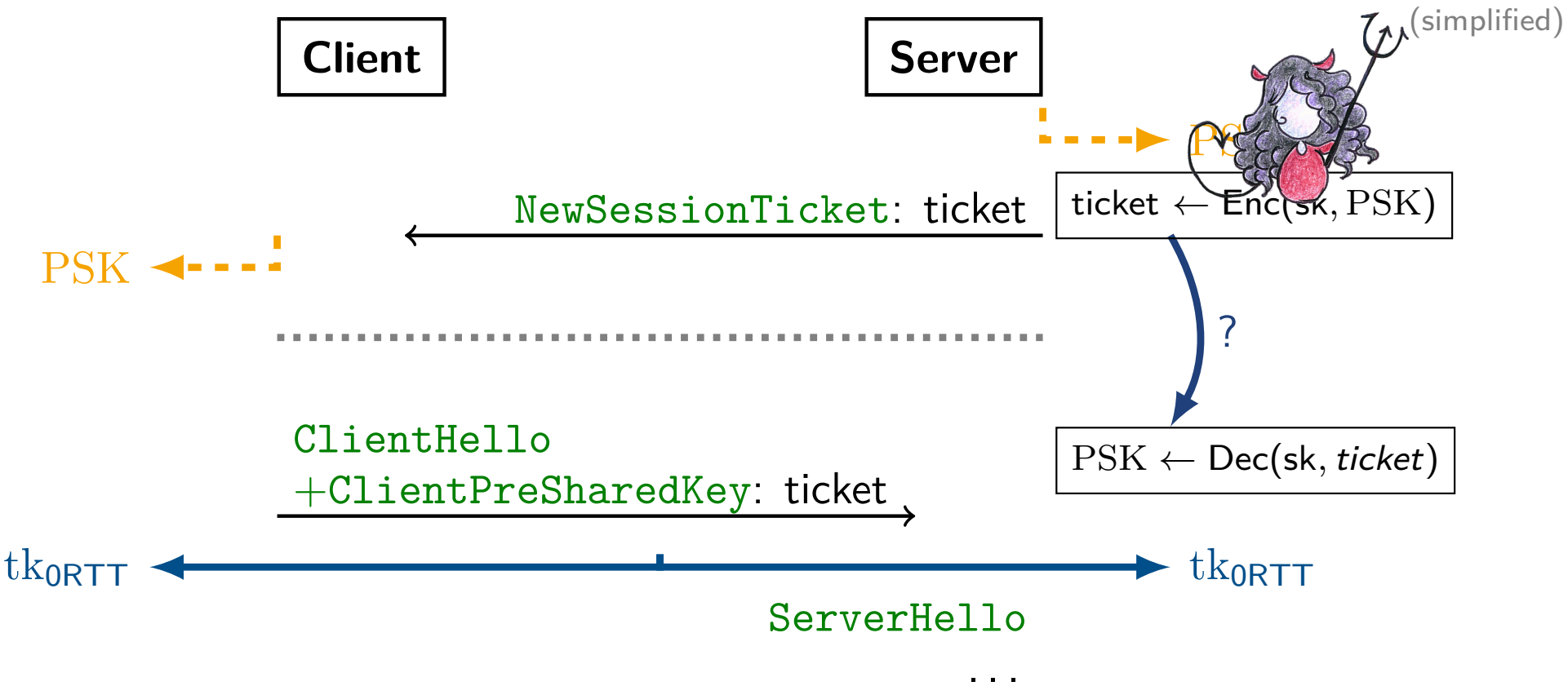
ETH zürich

(simplified)



The TLS 1.3 PSK/Resumption Handshake

ETH zürich



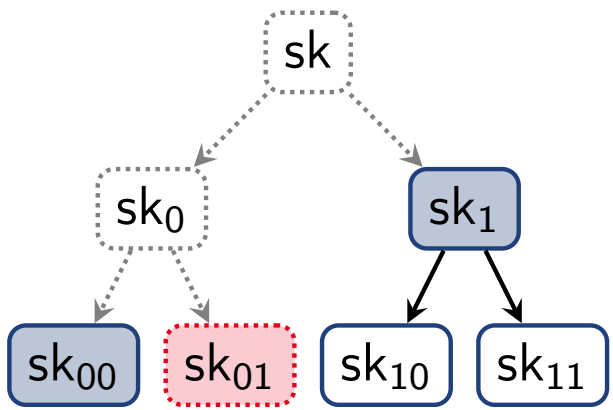
Puncturing to the rescue...

ETH zürich

- What if we could "forget" the ability to decrypt a ticket after using it?
- Aviram, Gellert, Jager (2019): "Session resumption protocols & fs TLS 1.3"
  - idea: use *sk* in **Puncturable** PRF + combine with AEAD

Puncturable PRF (PPRF)

- $\text{KeyGen}() \xrightarrow{\$} \text{sk}$
- $\text{Eval}(\text{sk}, x) \rightarrow y/\perp$
- $\text{Punc}(\text{sk}, x) \rightarrow \text{sk}'$



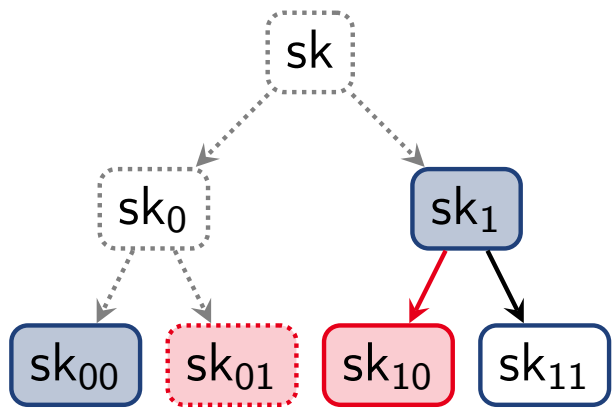
Puncturing to the rescue...

ETH zürich

- What if we could "forget" the ability to decrypt a ticket after using it?
- Aviram, Gellert, Jager (2019): "Session resumption protocols & fs TLS 1.3"
  - idea: use *sk* in **Puncturable** PRF + combine with AEAD

Puncturable PRF (PPRF)

- KeyGen() $\xrightarrow{\$}$ sk
- Eval(sk, x) $\rightarrow y/\perp$
- Punc(sk, x) $\rightarrow$ sk$'$



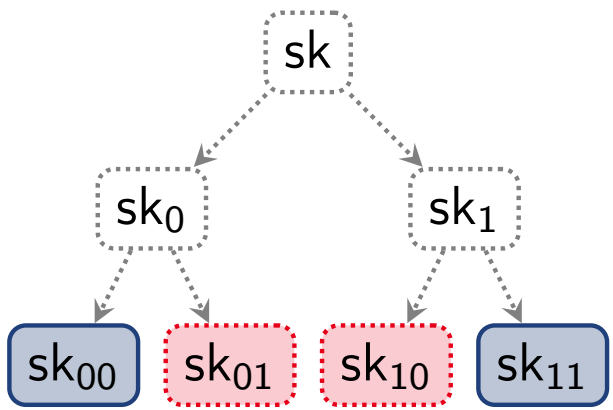
Puncturing to the rescue...

ETH zürich

- What if we could "forget" the ability to decrypt a ticket after using it?
- Aviram, Gellert, Jager (2019): "Session resumption protocols & fs TLS 1.3"
  - idea: use *sk* in **Puncturable** PRF + combine with AEAD

Puncturable PRF (PPRF)

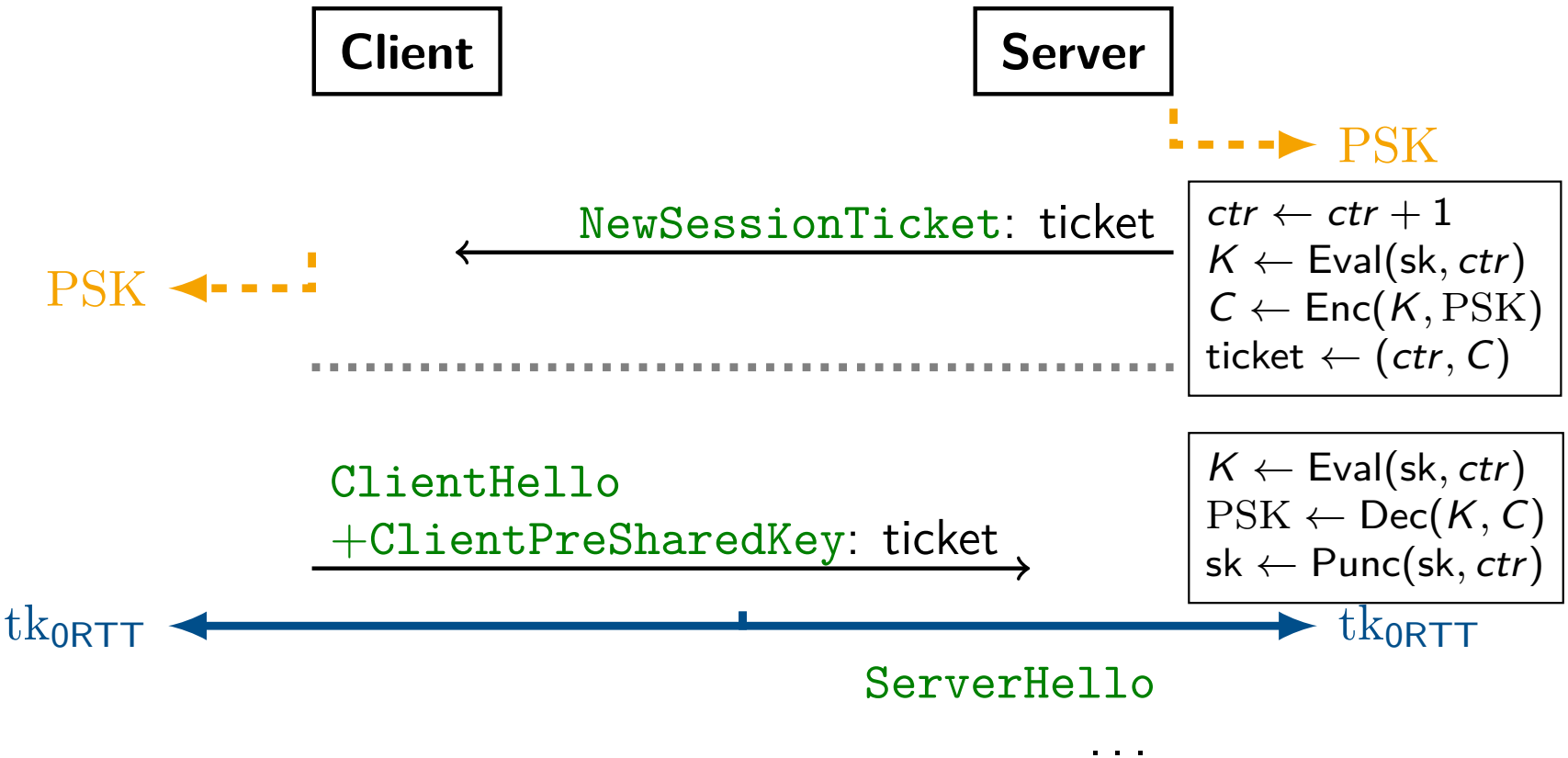
- $\text{KeyGen}() \xrightarrow{\$} \text{sk}$
- $\text{Eval}(\text{sk}, x) \rightarrow y/\perp$
- $\text{Punc}(\text{sk}, x) \rightarrow \text{sk}'$



Forward-secure Resumption from PPRF+AEAD [AGJ]

ETH zürich

(simplified)



- ✓ forward security: through PPRF puncturing
- ▶ meta data: tickets now possibly linkable, via counters
- ▶ is there a more general perspective?

Puncturable Key Wrapping

Syntax

ETH zürich

- idea: merge classical **key wrapping** (Rogaway-Shrimpton, 2006) with **puncturable encryption** (Green-Miers, 2015)

Puncturable Key Wrapping (PKW)

- KeyGen() $\xrightarrow{\$}$ sk
- Wrap(sk, T, H, K) $\rightarrow C/\perp$
- Unwrap(sk, T, H, C) $\rightarrow K/\perp$
- Punc(sk, T) $\rightarrow$ sk$'$

Remarks:

- puncturing on tags T, for now: think nonce
  but may subsume multiple ciphertexts/keys

PKW[PPRF,AEAD]

KeyGen():

1 Return PPRF.KeyGen()

Wrap(sk_{pprf}, T, H, K):

2 $sk_{aead} \leftarrow$ PPRF.Eval(sk_{pprf}, T)
3 $C \leftarrow$ AEAD.Enc(sk_{aead}, T, H, K)
4 Return C

Punc(sk_{pprf}, T):

8 $sk'_{pprf} \leftarrow$ PPRF.Punc(sk_{pprf}, T)
9 Return sk'_{pprf}

Puncturable Key Wrapping

Security

ETH zürich

- ex.: **confidentiality**
- we also define integrity and some further properties

Game $\mathbf{G}^{\text{find\$-cpa}}_{\text{PKW}}(\mathcal{A})$, $\boxed{\mathbf{G}^{\text{find\$-rcpa}}_{\text{PKW}}(\mathcal{A})}$:

1 $b \leftarrow\$ \{0,1\}$; $u \leftarrow 0$
2 $b^* \leftarrow\$ \mathcal{A}()$
3 Return $b^* = b$

NEW():

4 $u{+}{+}$
5 $sk_u \leftarrow\$ \mathsf{KeyGen}()$
6 $\mathcal{S}_{PT,u}, \mathcal{S}_{\$T,u}, \mathcal{S}_{T,u} \leftarrow \emptyset$
7 $\mathsf{corr}_u \leftarrow \text{false}$

WRAP(i, T, H, K):

8 If $T \in \mathcal{S}_{T,i}$ then return $\perp$
9 $C \leftarrow \mathsf{Wrap}(sk_i, T, H, K)$
10 $\mathcal{S}_{T,i} \xleftarrow{\cup} \{T\}$
11 Return C

RO\$-WRAP(i, T, H, K):

12 If $T \in \mathcal{S}_{T,i}$ or corr_i:
13 Return $\perp$
14 $C_1 \leftarrow \mathsf{Wrap}(sk_i, T, H, K)$
15 If $C_1 = \perp$ then return $\perp$
16 $C_0 \leftarrow\$ \{0,1\}^{\mathsf{cl}(|K|)}$
17 $\mathcal{S}_{\$T,i} \xleftarrow{\cup} \{T\}$; $\mathcal{S}_{T,i} \xleftarrow{\cup} \{T\}$
18 Return C_b

CORR(i):

19 If $\mathcal{S}_{\$T,i} \not\subseteq \mathcal{S}_{PT,i}$:
20 Return $\perp$
21 $\mathsf{corr}_i \leftarrow \text{true}$
22 Return sk_i

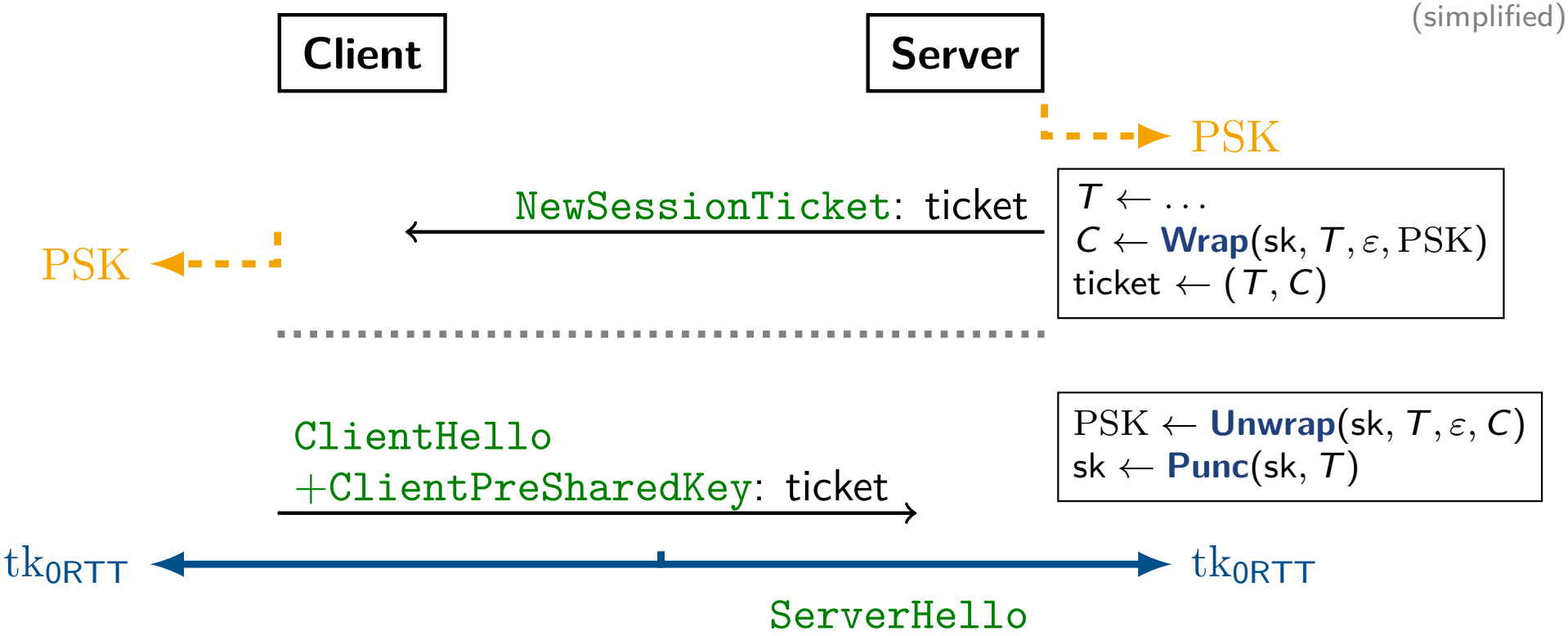
PUNC(i, T):

23 $sk_i \leftarrow \mathsf{Punc}(sk_i, T)$
24 $\mathcal{S}_{PT,i} \xleftarrow{\cup} \{T\}$

Forward-secure Resumption from PKW

ETH zürich

(simplified)



- generic, natural instantiation
- ✓ recast **[AGJ]**: take T to be a counter *ctr* ($\rightarrow$ identified gaps)
- ✓ meta-data **privacy**? sample T at random (req. larger space)
- more **coarse-grained** forward security? reuse T across tickets (req. MRAE)

Concluding Thoughts

ETH zürich

- **Puncturable Key Wrapping** as a conceptual abstraction for fine-grained forward security in symmetric key hierarchies
- **Constructions**:
  - generic PPRF+AEAD (w/ or w/o MR), others?
    - unified PPRF notions along the way
  - tag-based syntax allows for
    - ctr — possibly efficient puncturing
    - $ — enhanced privacy
    - middle ground(s)?
- **Applications:**
  - forward-secure resumption, rephrasing [AGJ]
  - cloud storage w/ fine-grained forward security
  - ...

Thank You!

mail@**felixguenther.info**