Alexander Haar

Key exchange using an online trusted third party

Outline

- overview
- TTP protocol
- different insecure Variotions
- conclusion

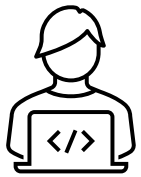
Overview

- TTP must play an active role in all exchanges
- efficient for clients
- heavy load on TTP
- usecase: corporate networks

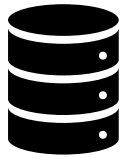
$\mathcal{K}_e$: Cipher key space

$\mathcal{K}_m$: MAC key space

Registration



Peter



TTP

(id_P)

$k_P \xleftarrow{R} (k_{enc,P},\ k_{mac,P}) \in \mathcal{K}_e \times \mathcal{K}_m$

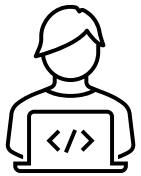
Stores only (id_P) for security reasons

(k_P)

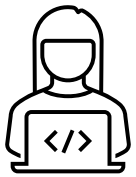
To Generate k_P again:

$k_P \leftarrow F(k_{TTP}, id_P)$

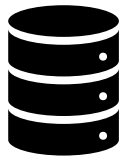
Key Exchange



Peter



Quinn



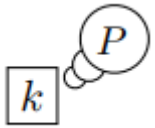
TTP

$r_P \xleftarrow{\$} \{0,1\}^*$

$r_Q \xleftarrow{\$} \{0,1\}^*$

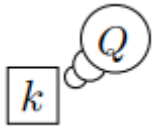
(id_P , r_P)

(r_P , r_Q , id_P, id_Q)



$c_Q := Enc_Q(k)$,
$t_Q := Mac_Q(id_P , r_P, r_Q, c_Q)$

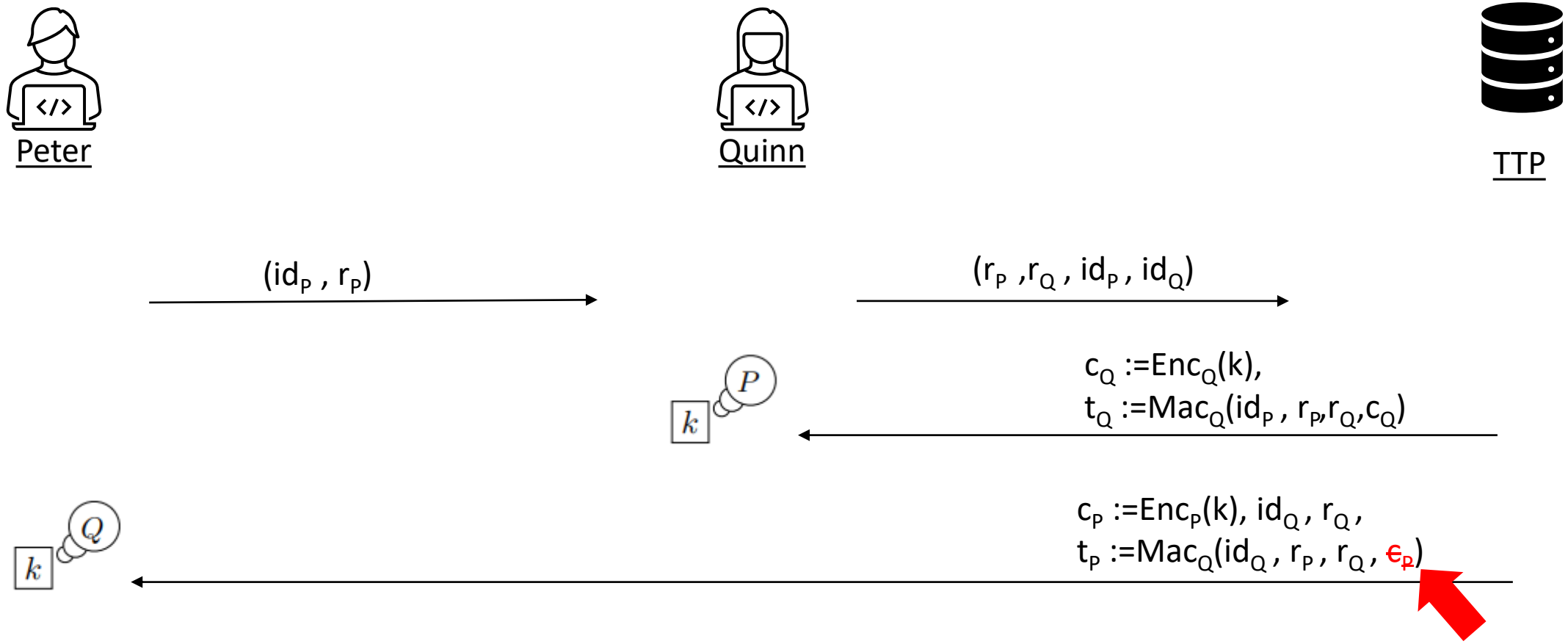
$c_P := Enc_P(k), id_Q , r_Q ,$
$t_P := Mac_Q(id_Q , r_P , r_Q , c_P)$



Insecure Variations

I. **key exposure attack**
II. replay attack
III. identity missbinding attack
IV. secure channel bindings attack

KEA – remove c_P from tag

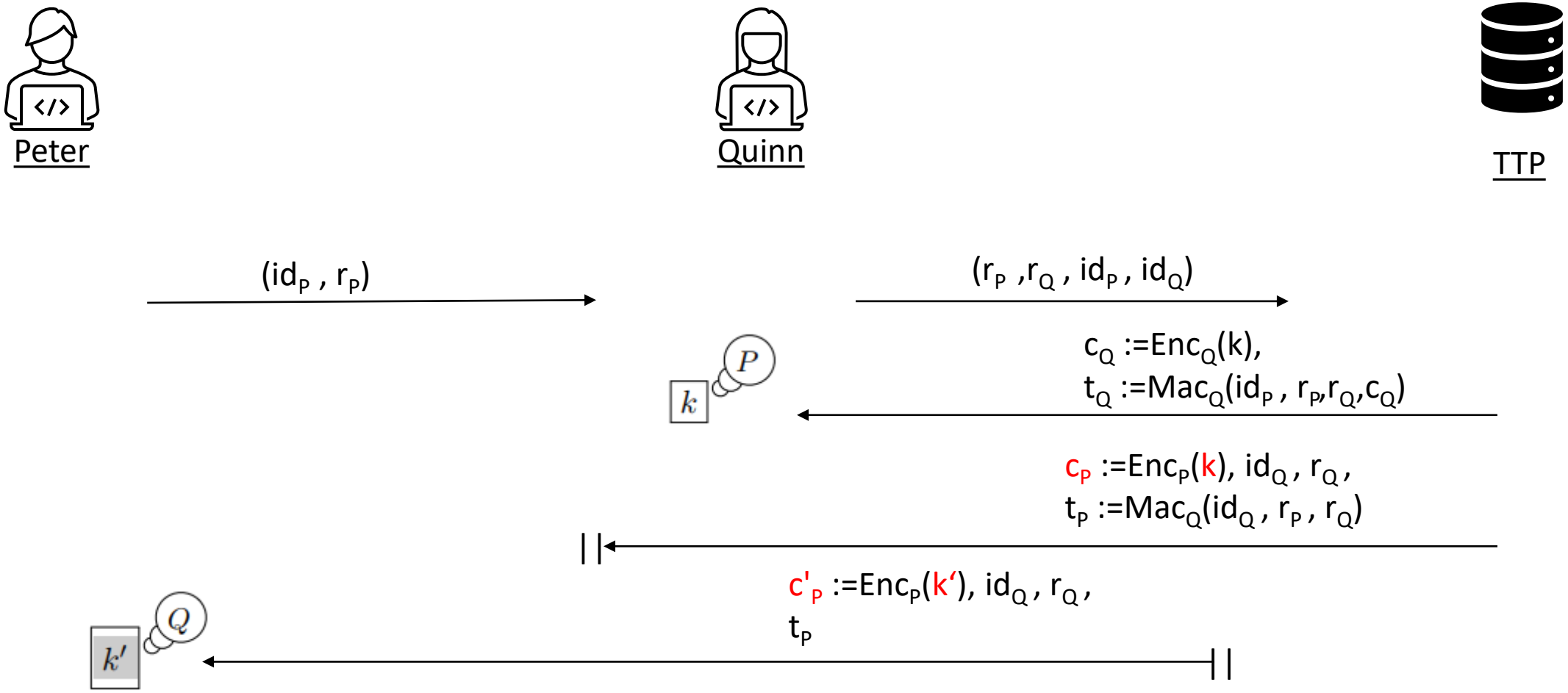


Key exposure attack

first step:

1. Adversary registers new user R
2. Initialize a conversation with Peter
3. Obtains $c_R := Enc_R(k')$ and decrypts it to obtain k´

KEA – second step



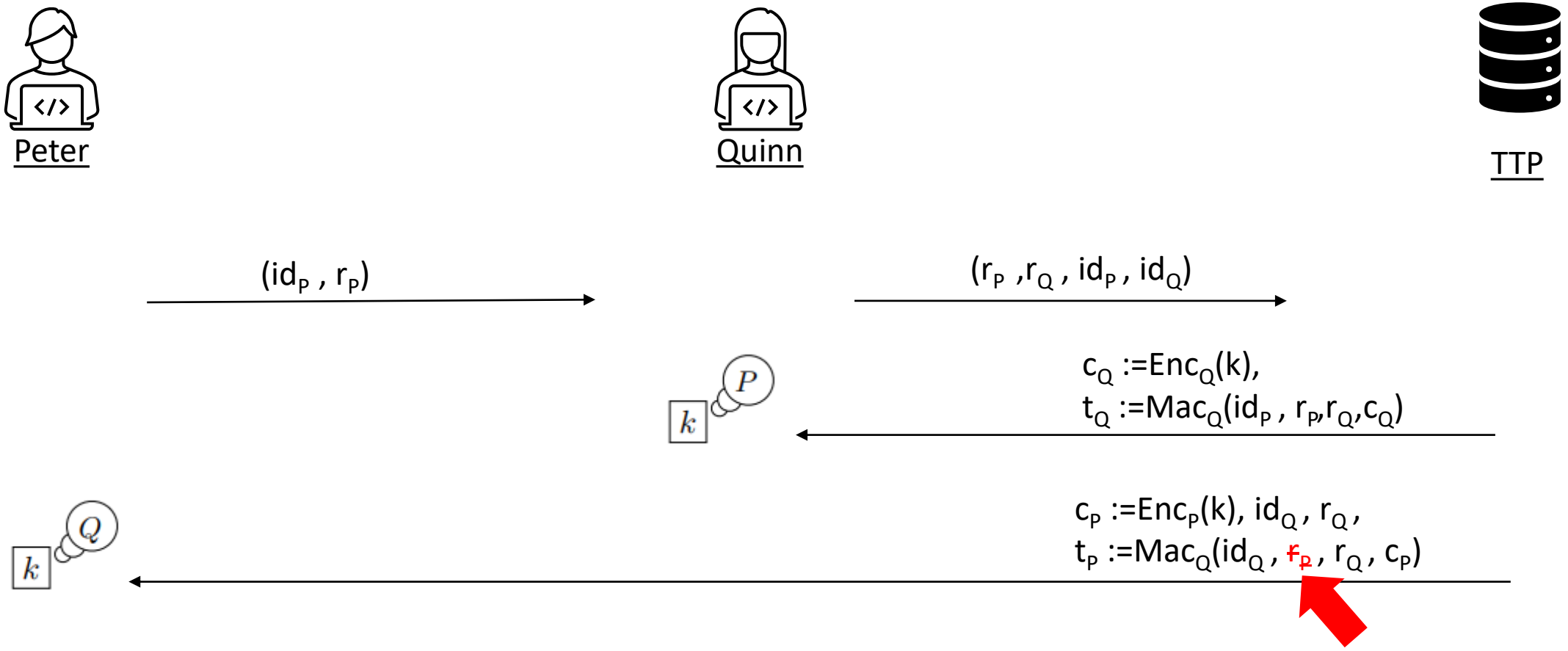
Key exposure attack

- adversary can now read every message from Peter
- same attack can be used against Quinn
  - Replace $\mathbf{c_Q}$ where $\mathbf{c'_Q} := Enc_Q(k')$

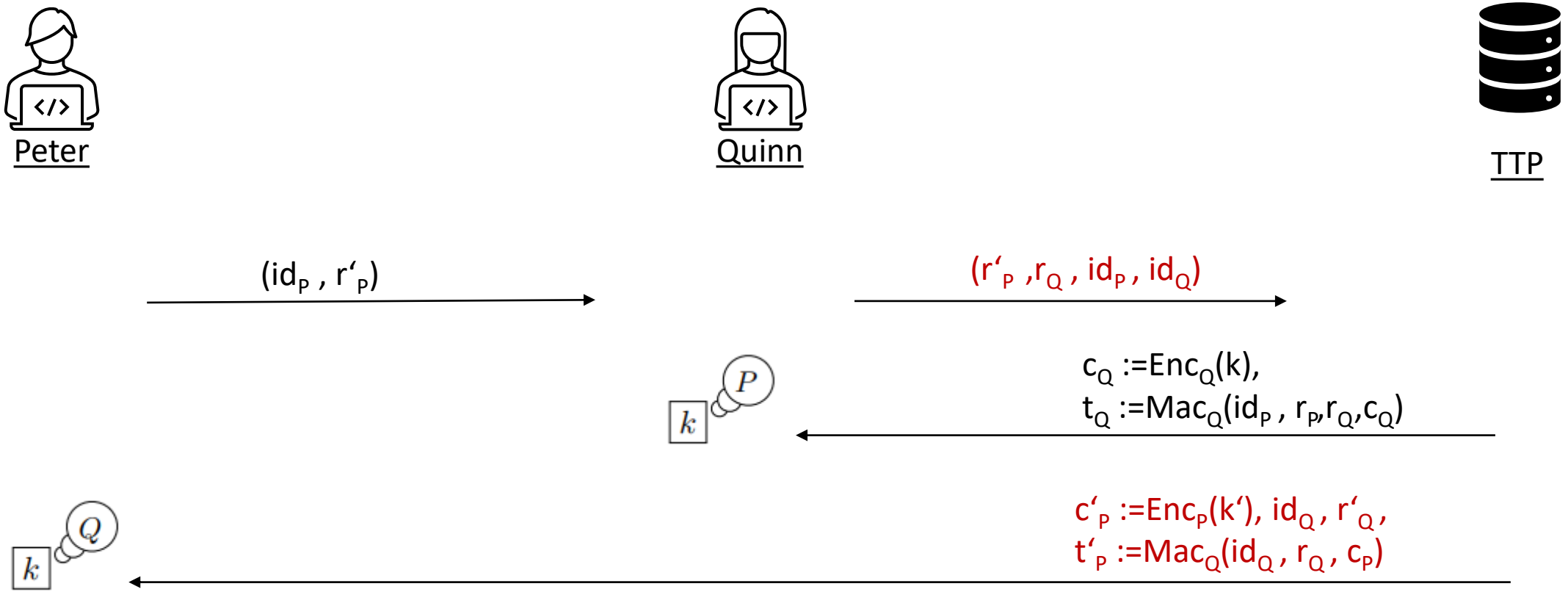
Insecure Variations

I. key exposure attack
II. replay attack
III. identity missbinding attack
IV. secure channel bindings attack

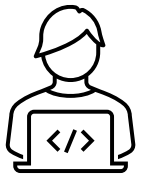
RA – remove r_P from tag



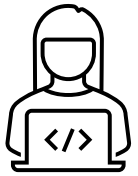
RA – remove r_P from tag



RA – remove r_P from tag



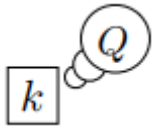
Peter



Quinn

(id_P , r_P)

||

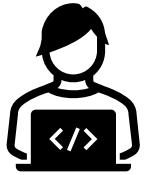


$c'_P := Enc_P(k'), id_Q , r'_Q ,$
$t'_P := Mac_Q(id_Q , r_Q , c_P)$

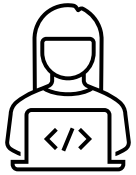
||

Peter thinks he is talking to Quinn
but he is holding the old session key k‘

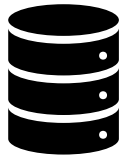
RA – attack on Quinn



Adversary



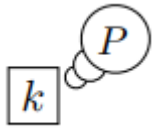
Quinn



TTP

(id_P , r'_P)

$(r'_P , r_Q , id_P , id_Q)$



$c_Q := Enc_Q(k)$,
$t_Q := Mac_Q(id_P , r'_P , c_Q)$
||

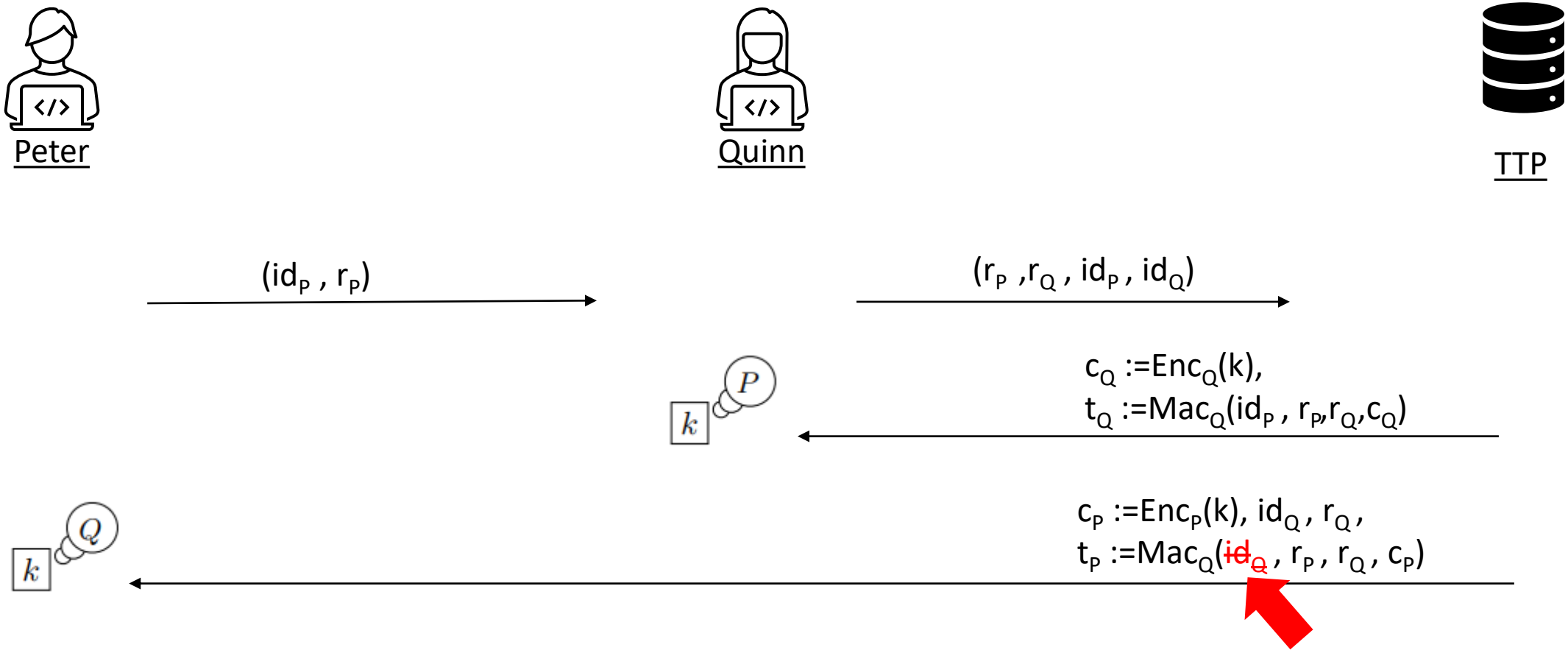
$c'_Q := Enc_Q(k')$,
$t'_Q := Mac_Q(id_P , r'_P , c_Q)$
||

Quinn thinks she is talking to Peter
but is holding the old session key k‘

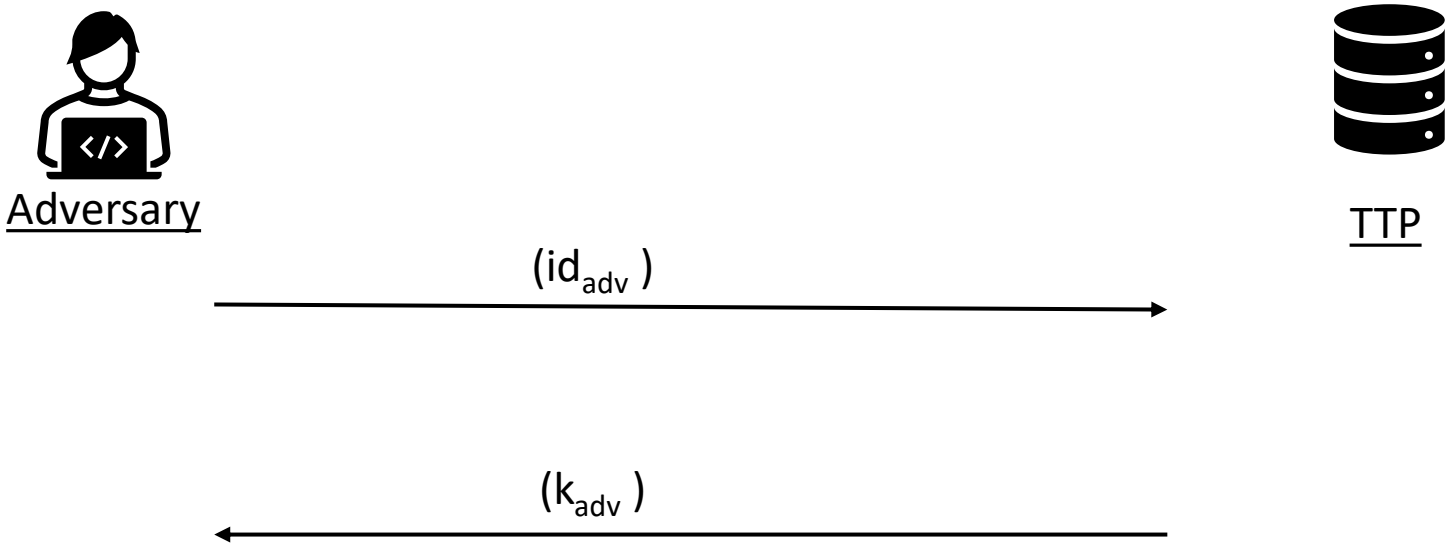
Insecure Variations

I. key exposure attack
II. replay attack
III. identity missbinding attack
IV. secure channel bindings attack

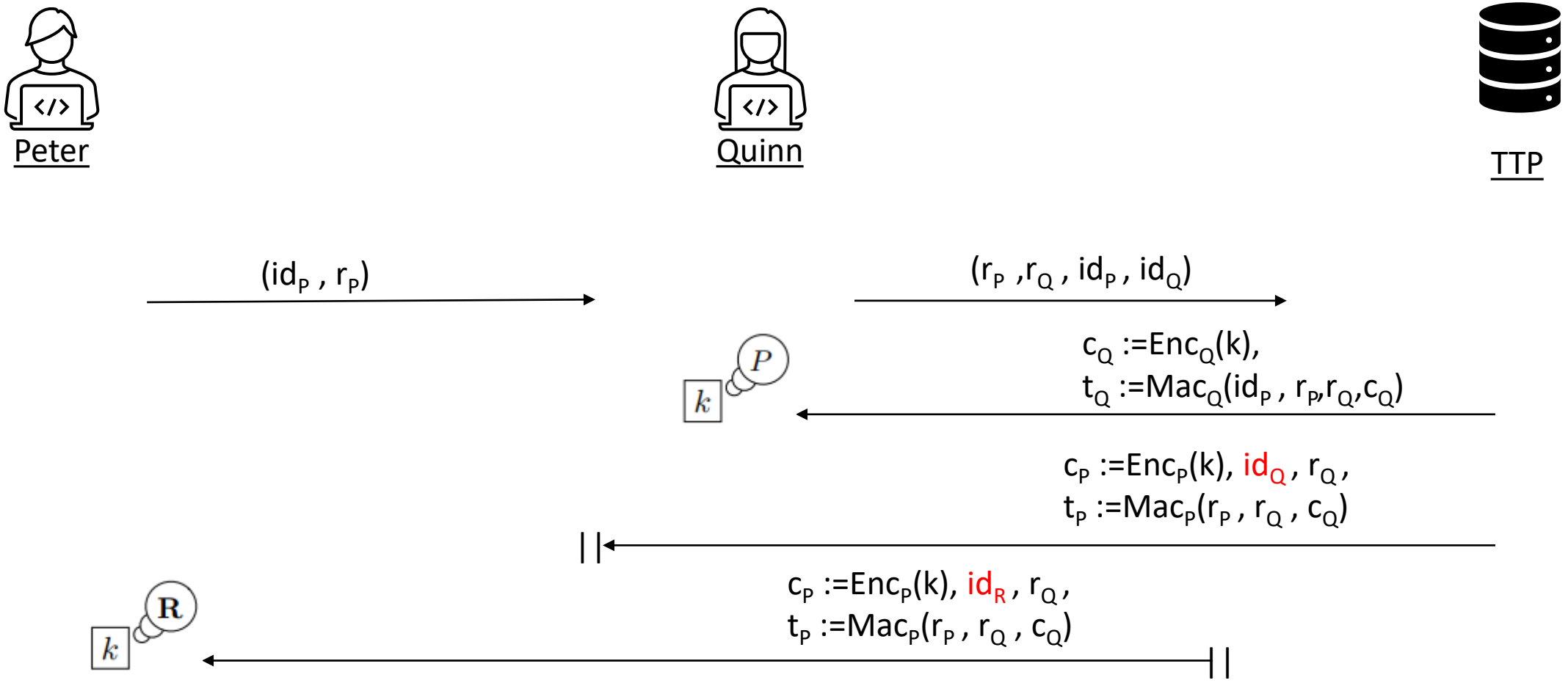
IMA – remove id_Q from tag



Adversary Registration



IMA – remove id_Q from tag

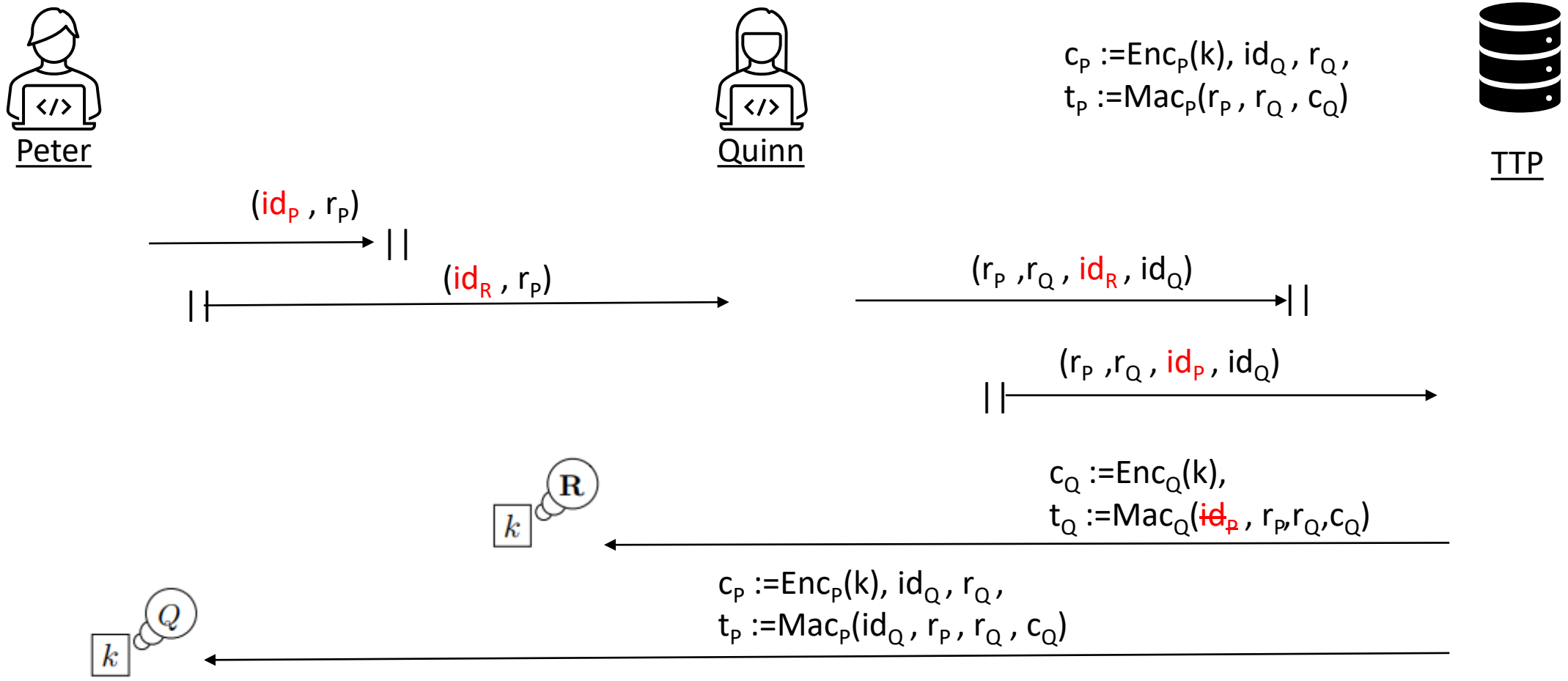


IMA – remove id_Q from tag

- both user share the same key k
- Adversary has no information on k

=> parties are misbound

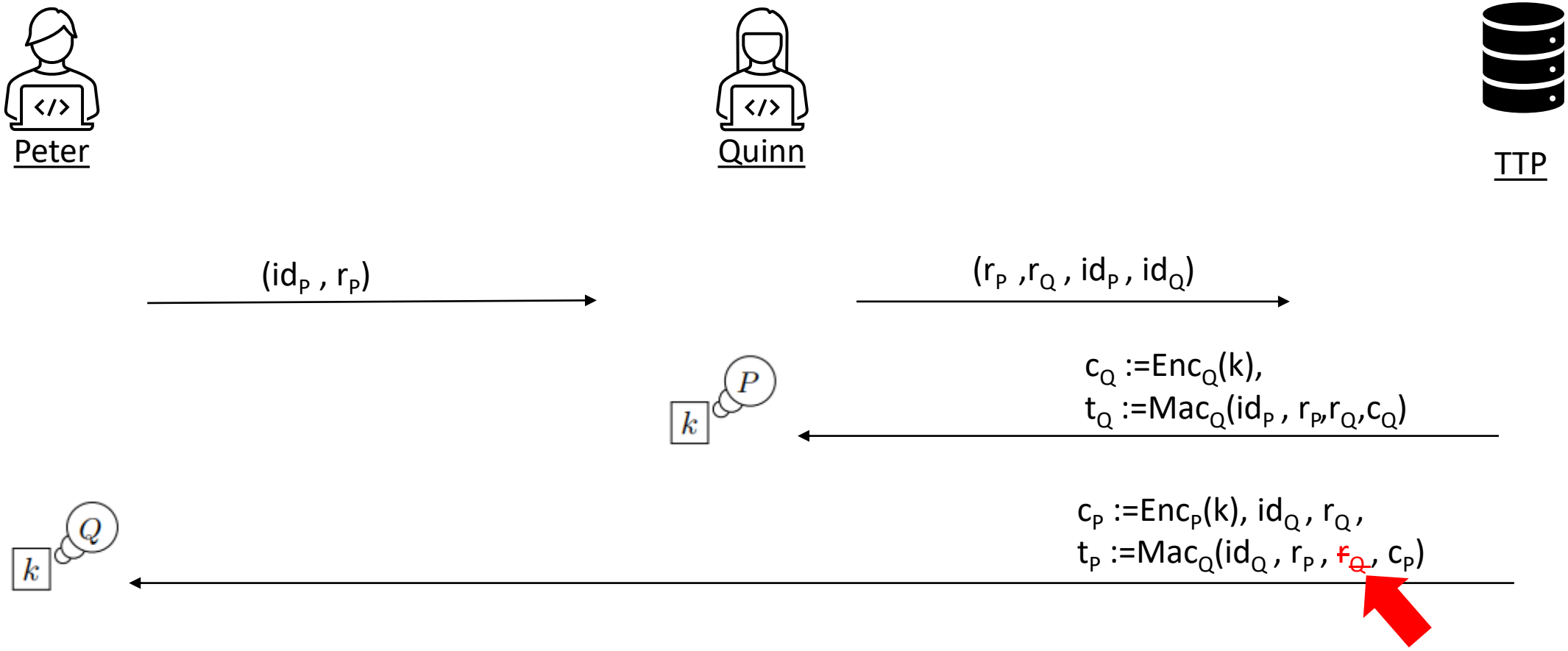
IMA – attack on Quinn



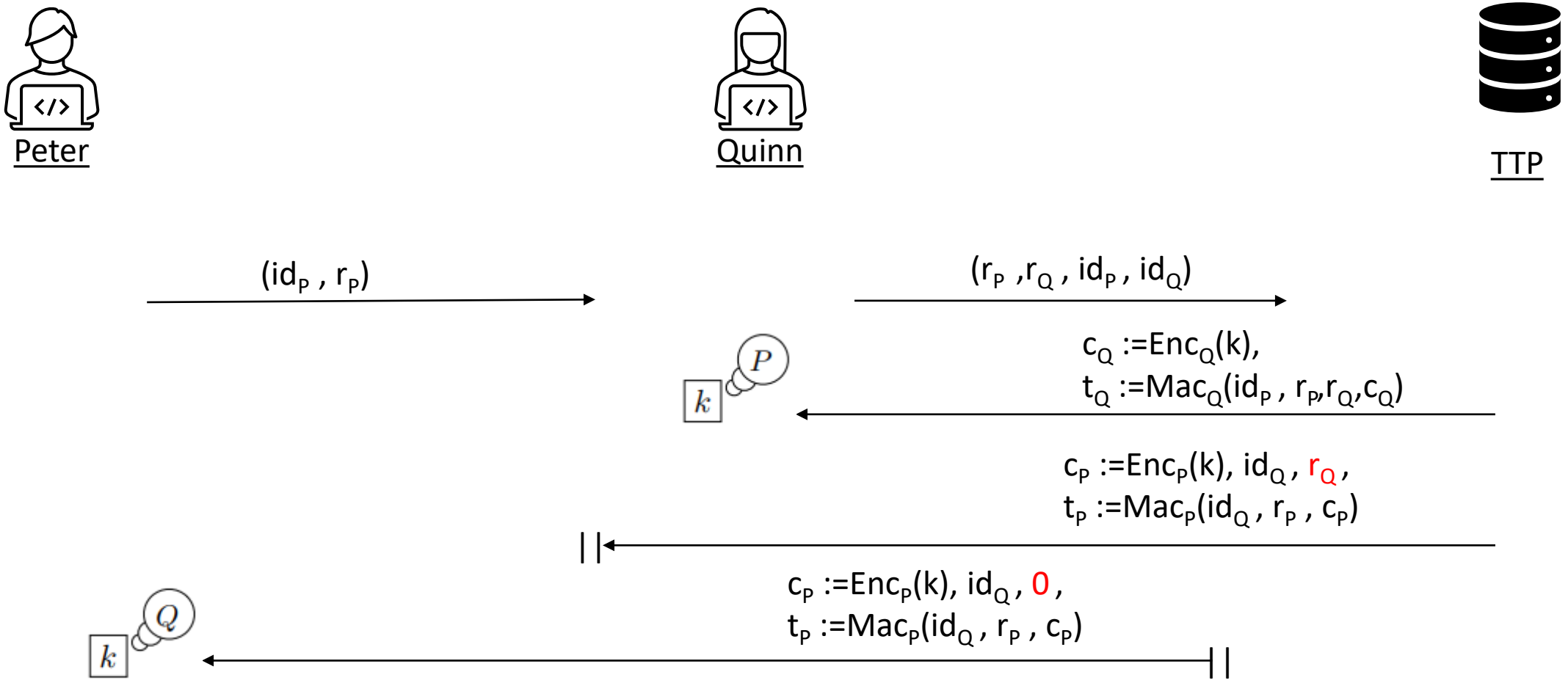
Insecure Variations

I. key exposure attack
II. replay attack
III. identity missbinding attack
IV. secure channel bindings attack

SCBA – remove r_Q from tag



SCBA – remove r_Q from tag



SCBA – remove r_Q from tag

- both sides agree on session key k
- k is unknown to the adversary
- disagree on their channel bindings

Conclusion

- TTP is a statically secure key exchange protocol
- can provide secure channel bindings
- Not PFS secure
  - If adv. learns either P's key, Q's key, or the TTP's key
  - All past sessions between P and Q are exposed

Ressource

- Dan Boneh and Victor Shoup „A Graduate Course in Applied Cryptography” Version 0.6, Jan 2023 https://crypto.stanford.edu/~dabo/cryptobook/BonehShoup_0_6.pdf