Analysis of the network security of the Estonian Mobile-ID identification protocol

Peeter Laud & Meelis Roos
Cybernetica AS & Tartu University

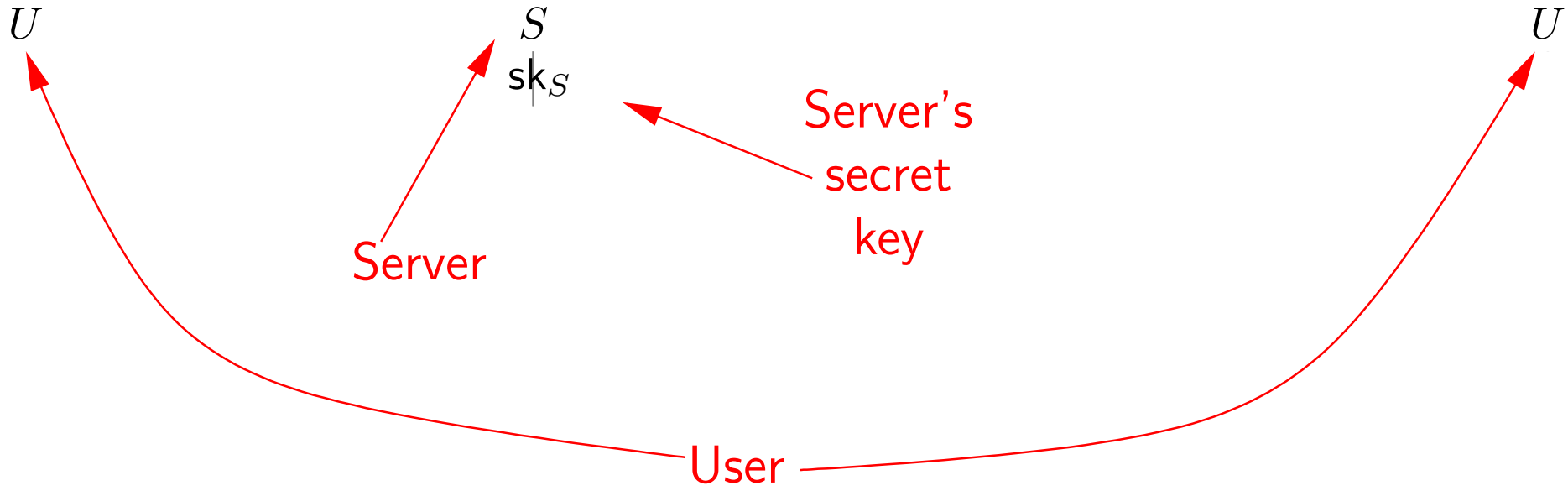
The object

- A SIM-card that
  - contains two private keys;
  - is capable of signing with those keys;
  - works like an "'ordinary"' SIM-card otherwise.
- During its activation SK AS issues certificates that
  - bind the corresponding public keys to your name;
  - state that the use of the first key is in identification
  - . . . and the use of the second key is in signing documents.

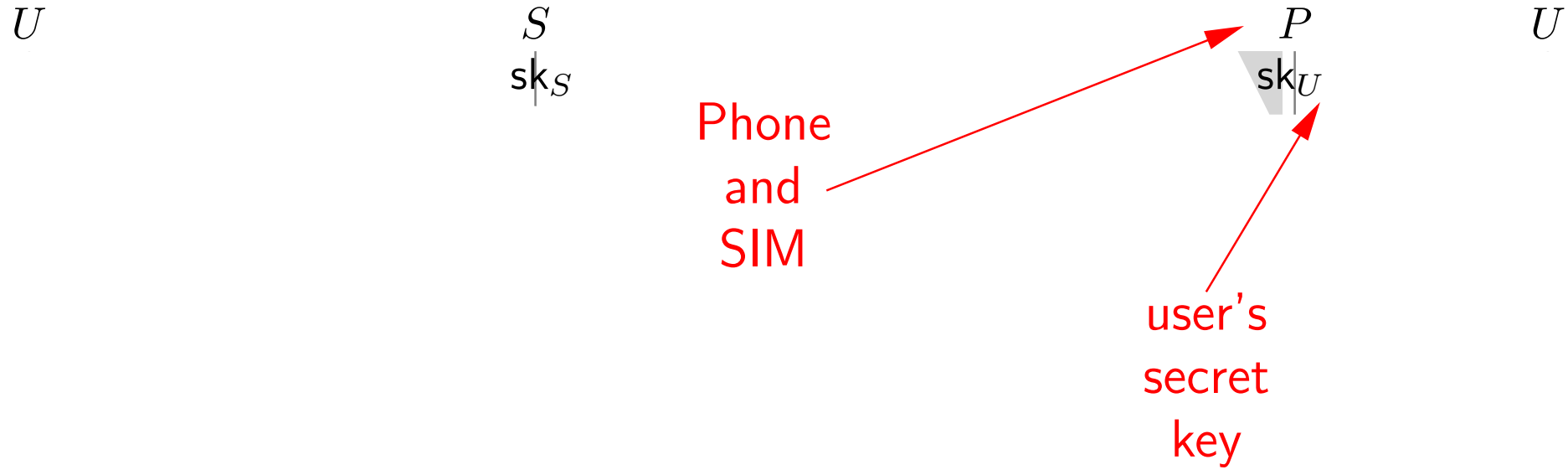
The signing procedure

- The card receives (x, M) from the mobile operator.
  - x — the (short) message to sign;
    - a couple of dozen bytes.
    - might be the hash of the "real" message.
  - M an explanatory text.
  - the channel from operator to SIM-card is secure.
- The card computes the control code $\mathrm{cc}(x)$ of x.
  - $\mathrm{cc}(x) \in \{0000, 0001, 0002, \ldots, 9999\}$
- The card shows $\mathrm{cc}(x)$ and M to the user (through the phone).
- If $\mathrm{cc}(x)$ and M OK, the user gives his/her PIN to the card.
  - Different PIN-s for different keys.
- The card verifies PIN, sends $\mathrm{sig}_{\mathsf{sk}}(x)$ to the operator.

The identification protocol



The identification protocol

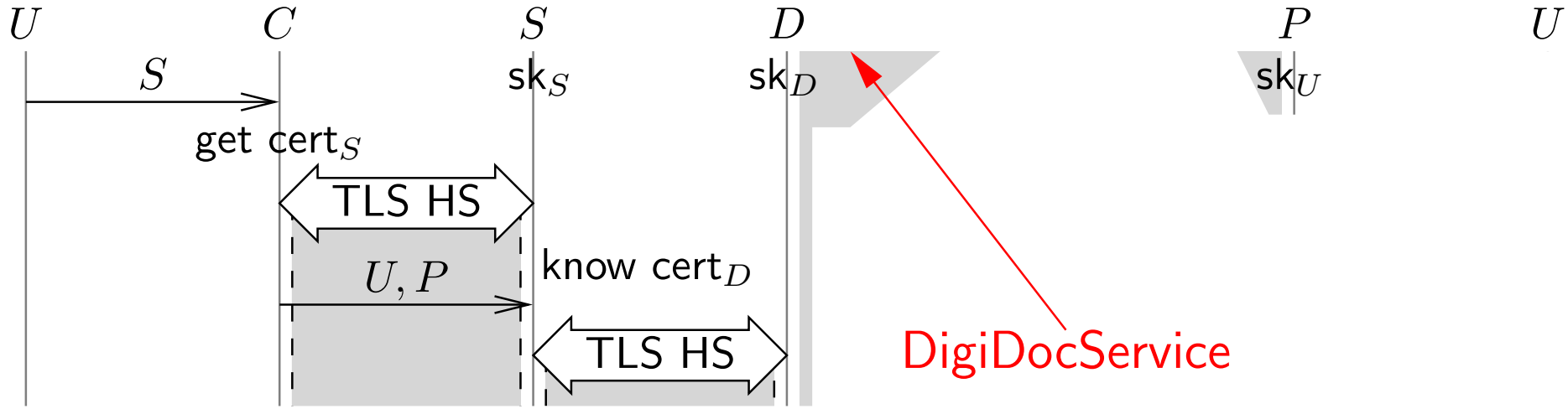


The identification protocol

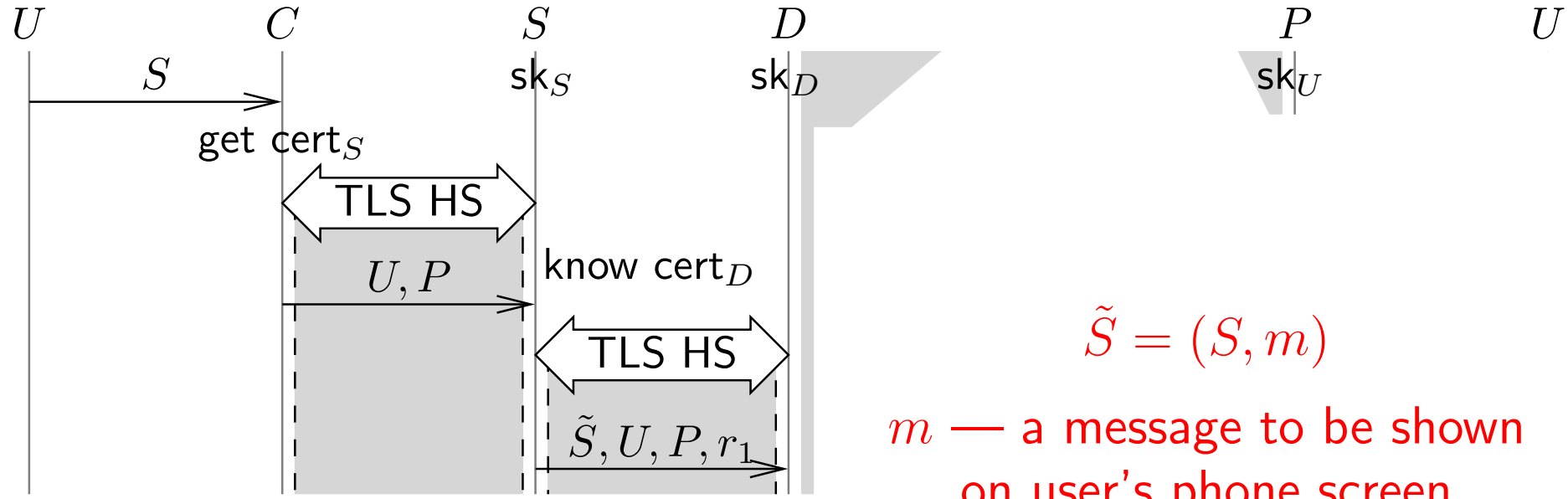
The identification protocol

The identification protocol

The identification protocol



The identification protocol

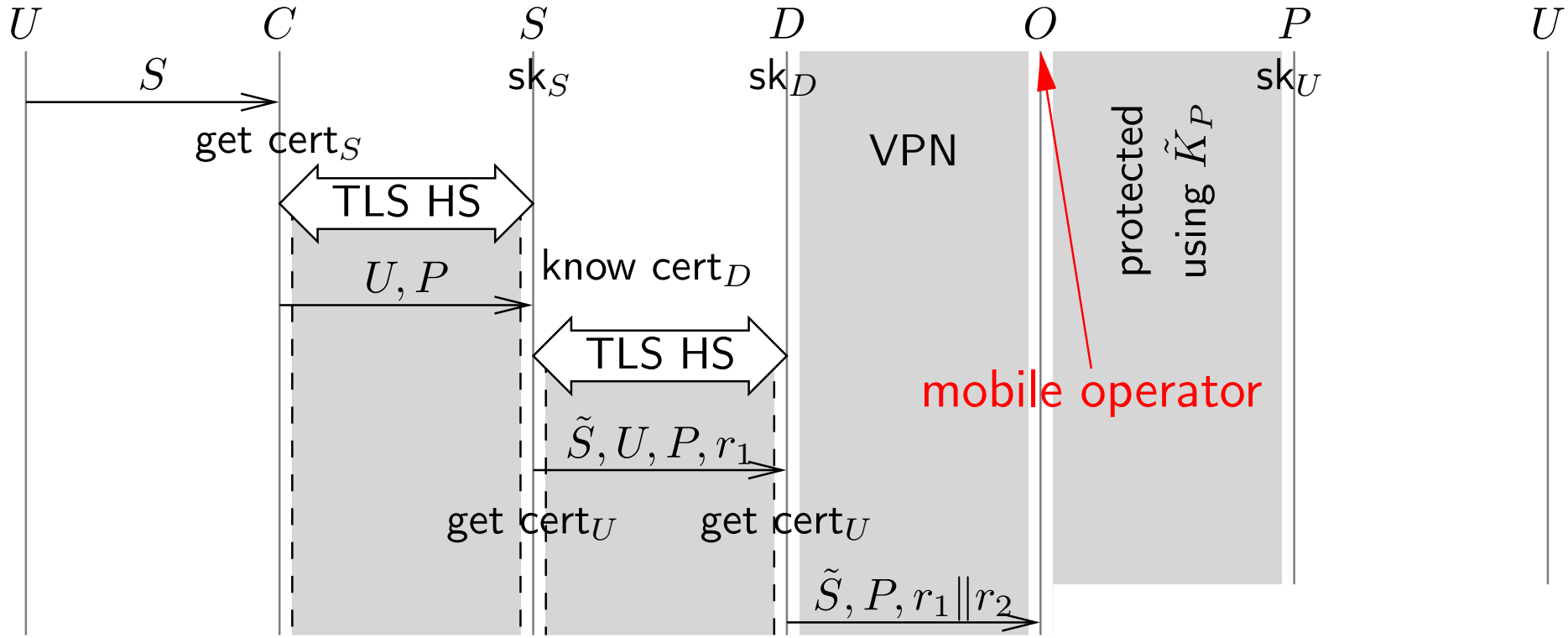


$\tilde{S} = (S, m)$

m — a message to be shown on user's phone screen

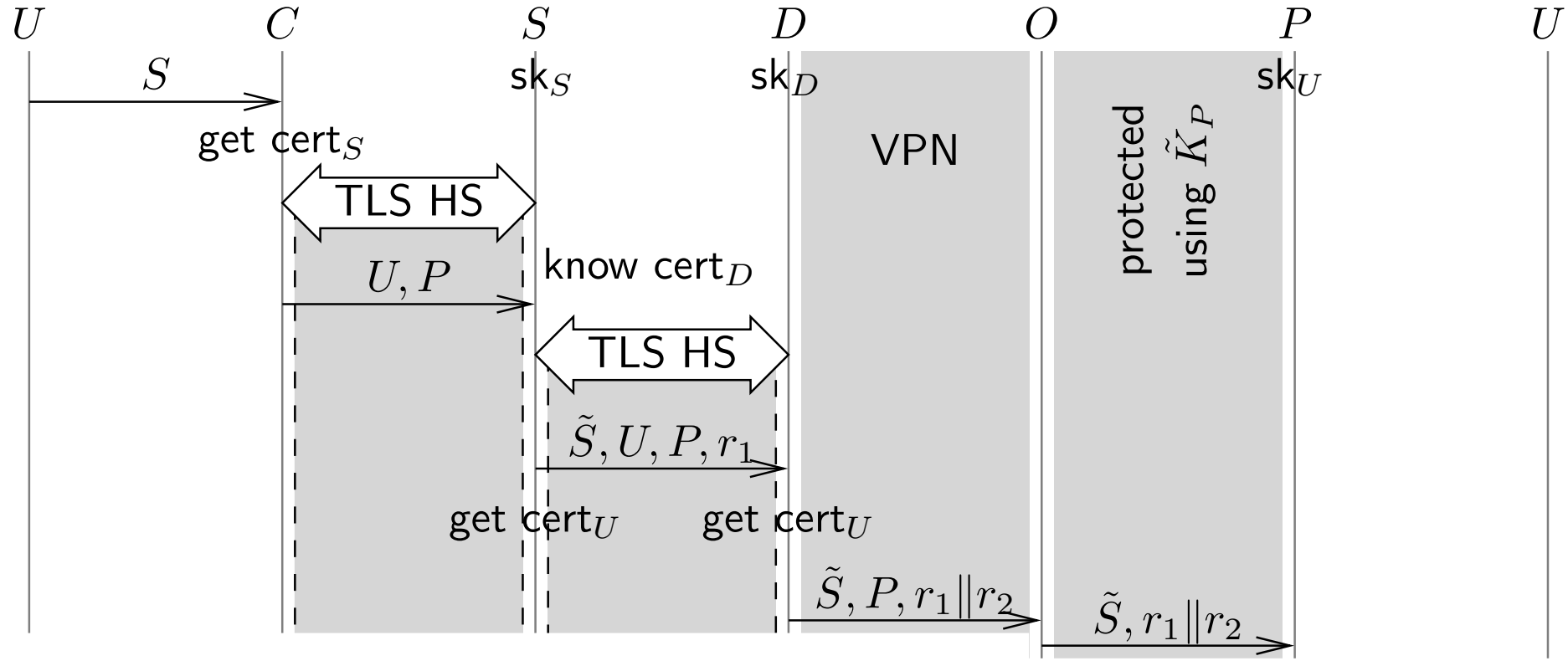
r_1 — a random number (10 bytes)

The identification protocol

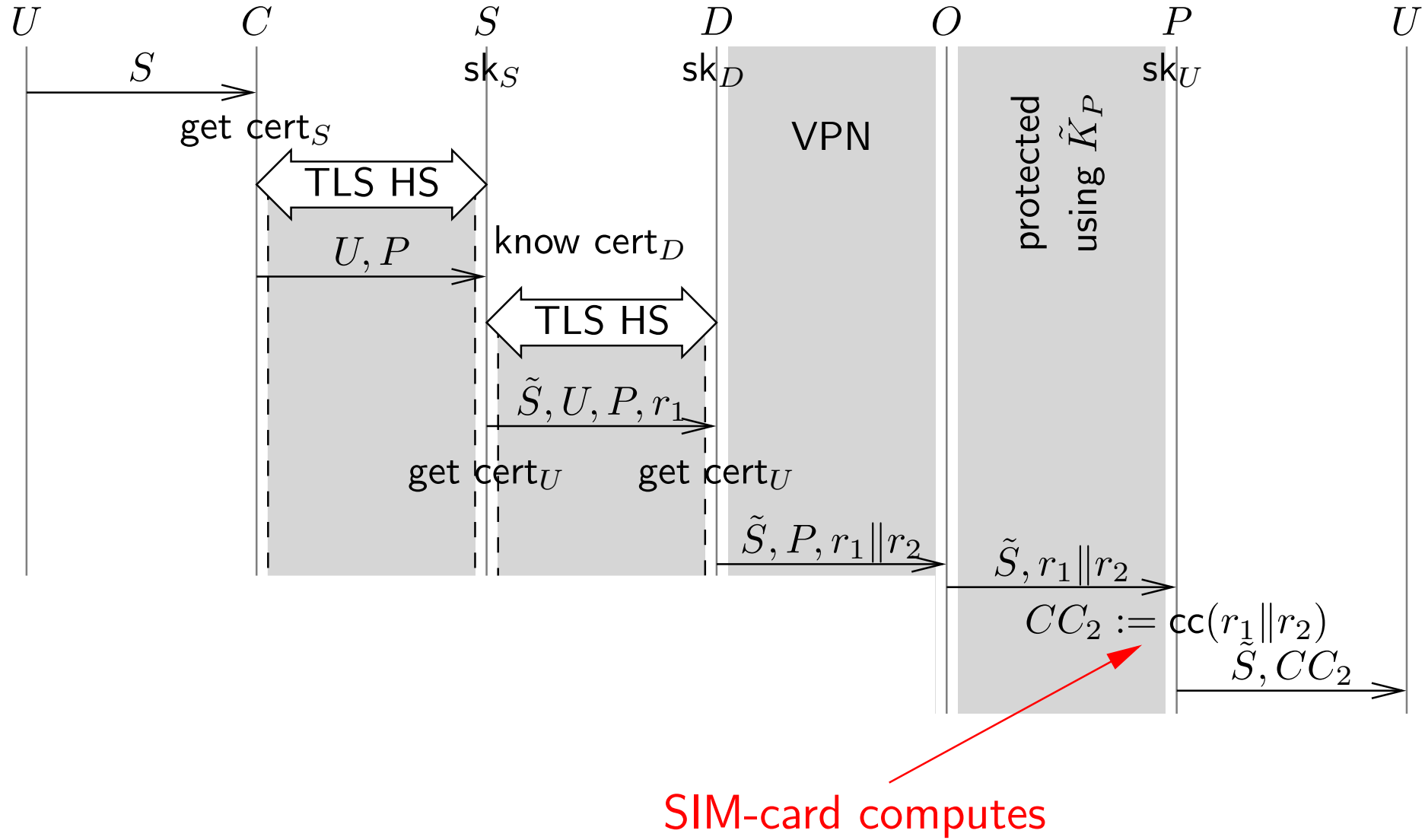


r_2 — a short random number

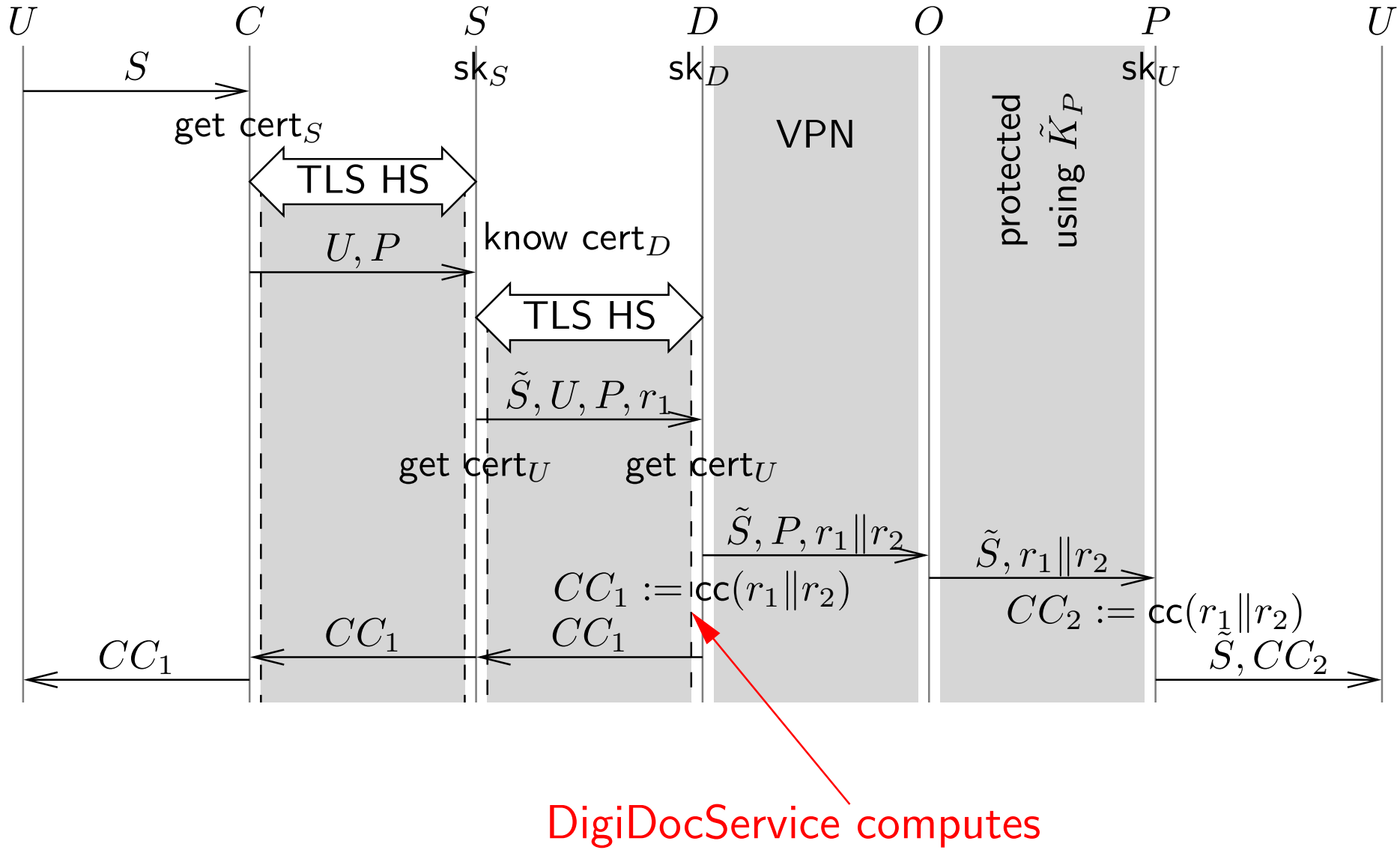
The identification protocol



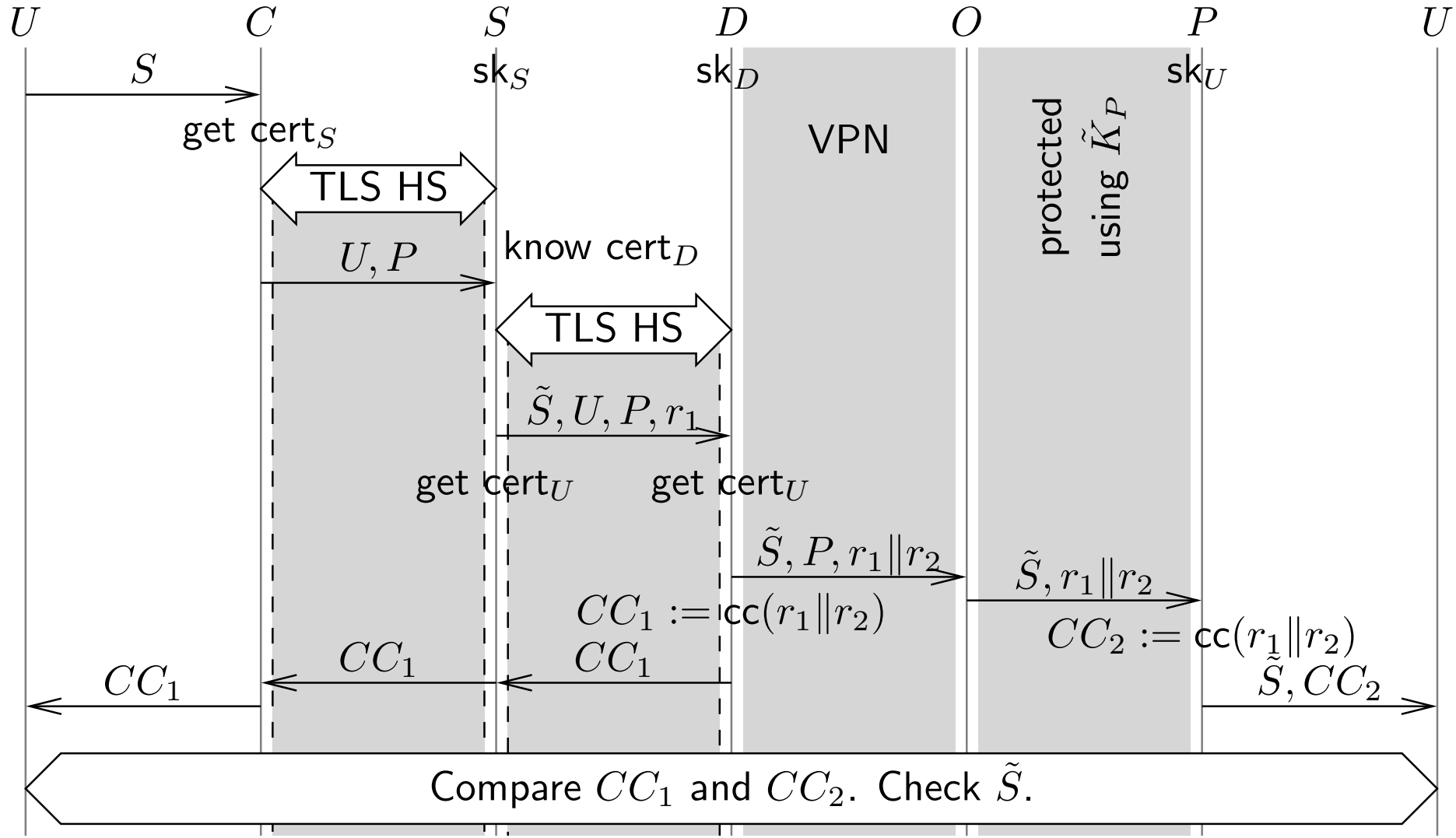
The identification protocol



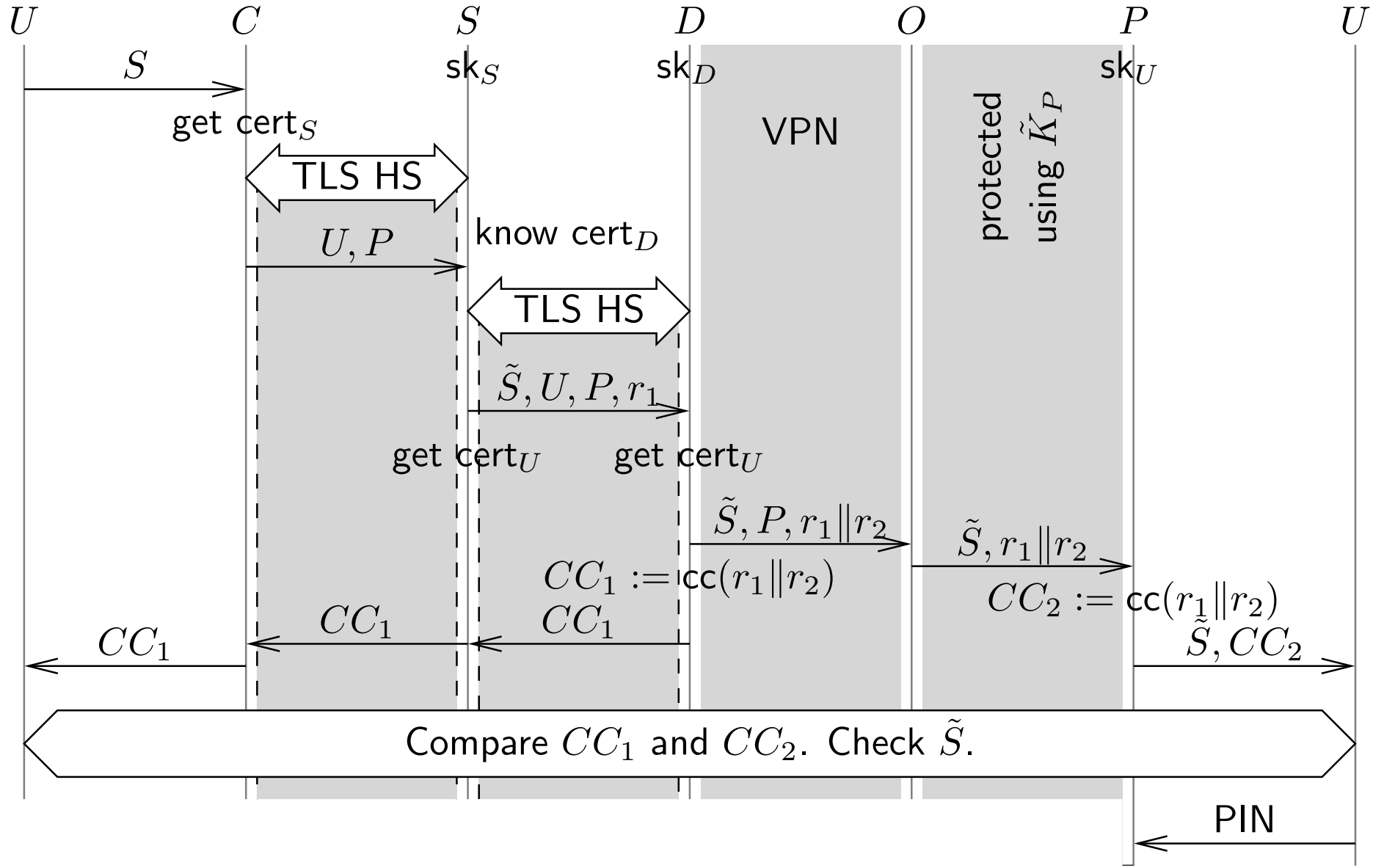
The identification protocol



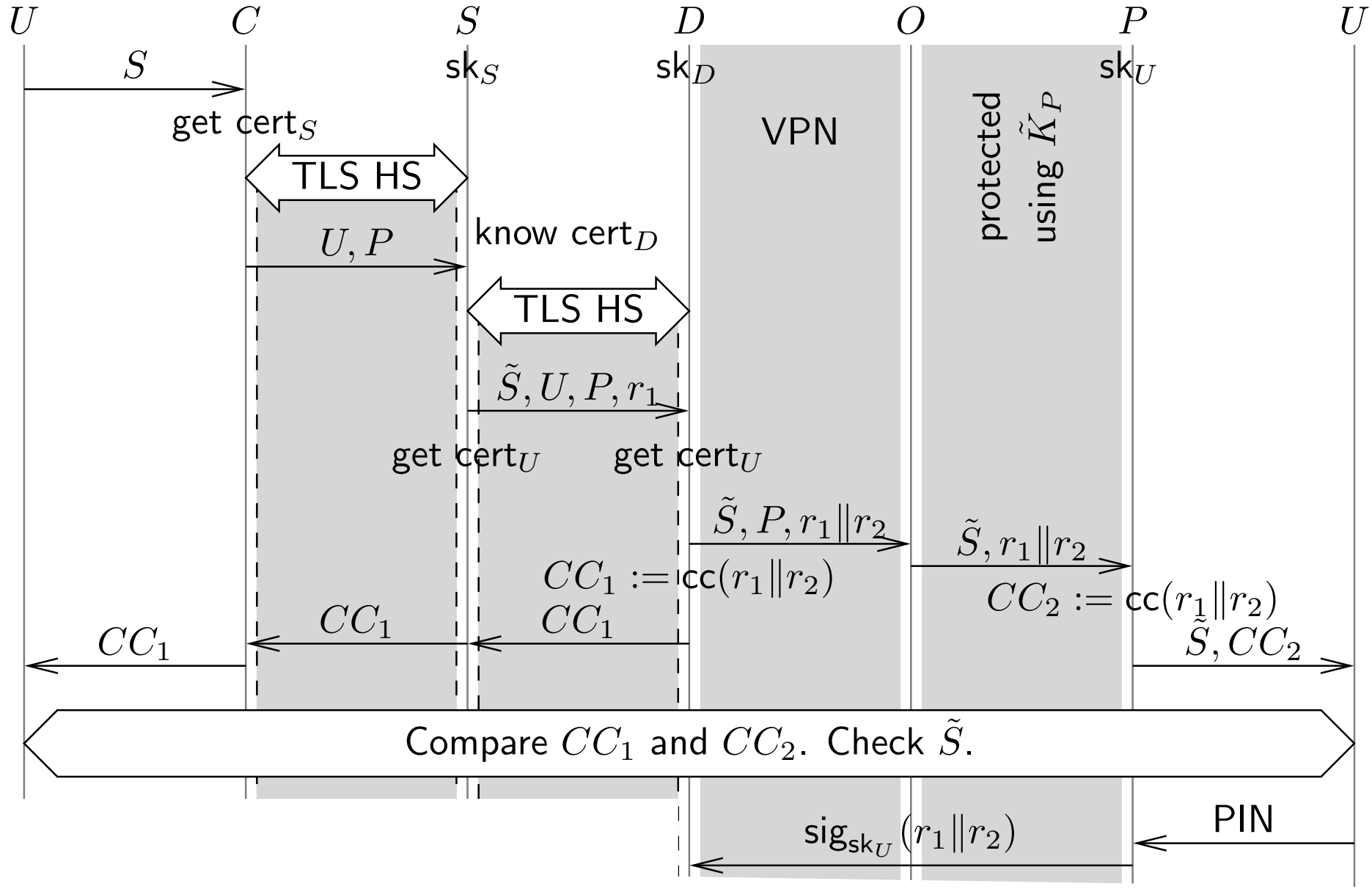
The identification protocol



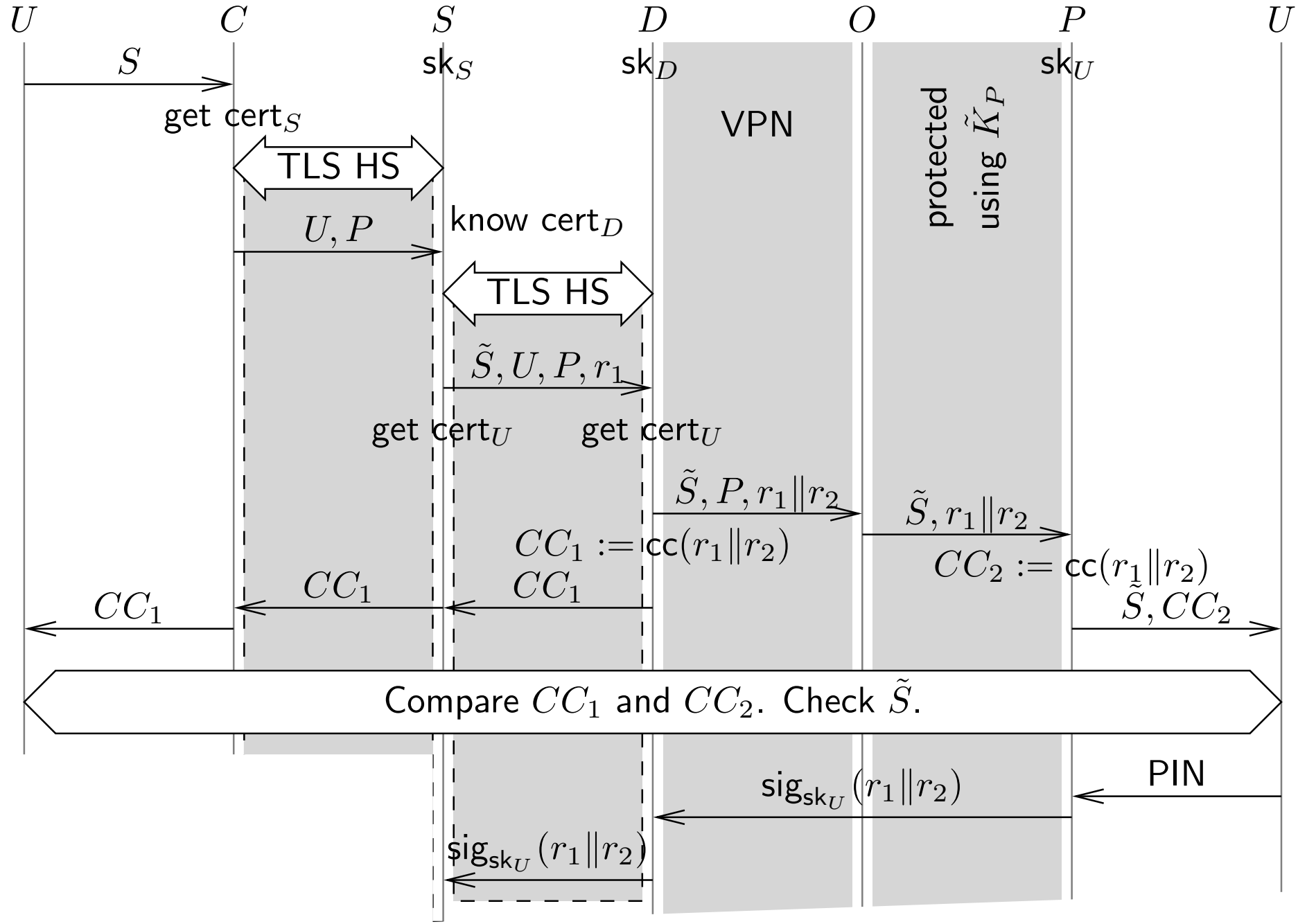
The identification protocol



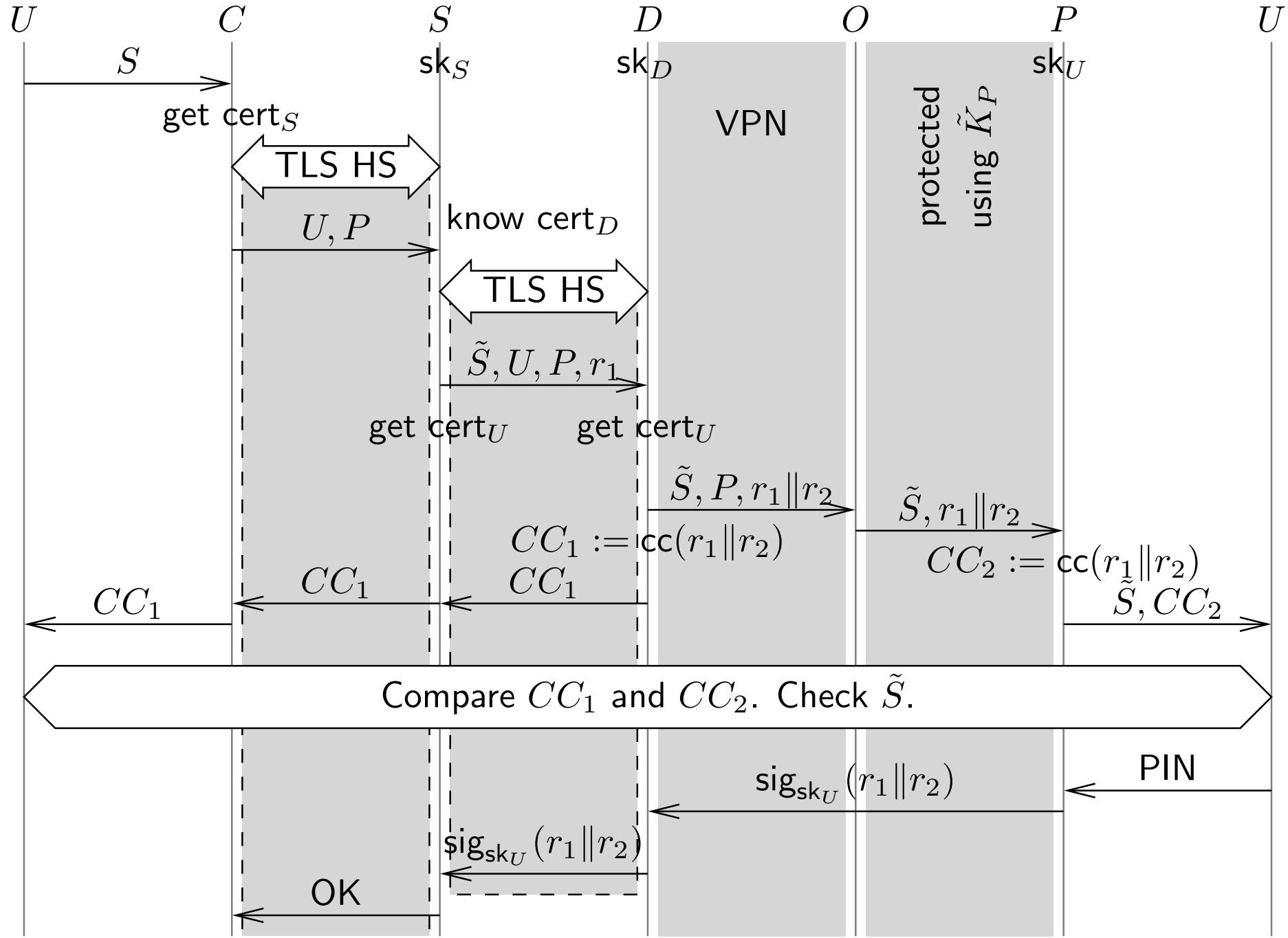
The identification protocol



The identification protocol



The identification protocol



“Base” security model

- There are several users and servers, some under adversarial control.
- DigiDocService and mobile operator are honest.
  - No confusion between different mobile operators.
- Client apps. and phones have no malware.
  - The channels between the user and client app. / phone are secure.
- The adversary controls the insecure channels. It can read and write them.
- The adversary can take messages apart and construct new messages. It can generate new keys, random numbers, etc.
- The adversary can start new sessions.
- The adversary schedules all parties.

Perfect cryptography assumption

- Messages have structure
  - It is their syntax tree.
- A message can be analysed only according to its structure:
  - From (m_1, m_2) find m_1 and m_2.
  - From $\mathsf{enc}_k(m)$ and k find m.
  - etc.
- To construct a message, we need all of its parts:
  - Need sk and m to construct $\mathsf{sig}_{\mathsf{sk}}(m)$.
  - etc.
- Different structure $\Rightarrow$ different message.
  - does not apply to control codes.
- This is a constraint on the adversary!

Security properties we care about

- If U and S are honest then the TLS key they agreed on will not become known to the adversary.
- If S thinks it talks to U using key K and U is honest then U thinks it talks to S using key K.

We are protecting an honest server

- Integrity for U follows from the properties of TLS handshake.

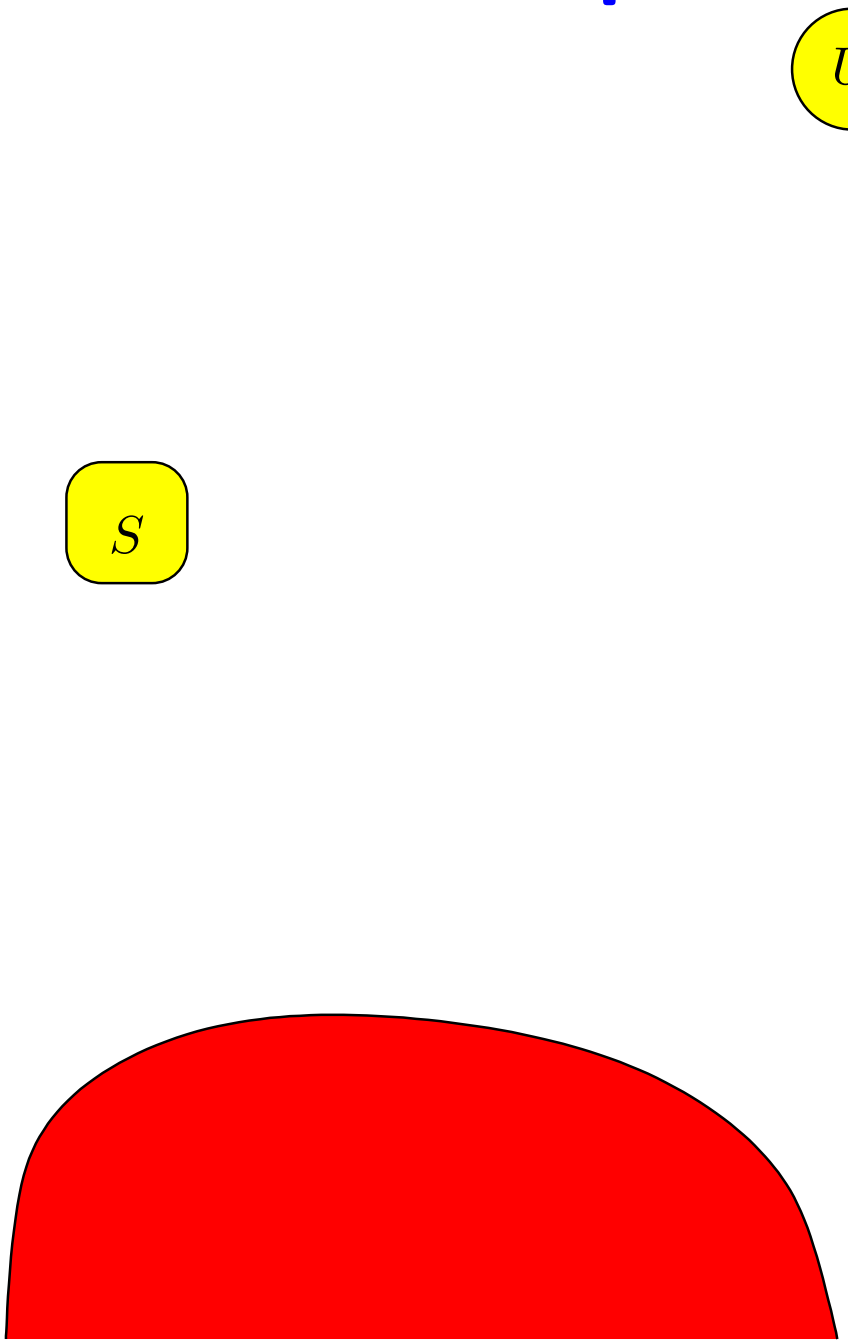
Analysing the protocol

- We use the perfect cryptography assumption.
- The question "does protocol $\mathfrak{P}$" satisfy the security property $\mathfrak{S}$?" is undecidable in general.
- Still, there are tools that take the description of a protocol and output whether it is secure.
  - Handle restricted classes of protocols.
  - Sometimes give wrong answer.
    - Only err at the side of caution.
- We have used ProVerif, `http://www.proverif.ens.fr`
- In the base security model the Mobile-ID identification protocol is **secure** against network attacks.

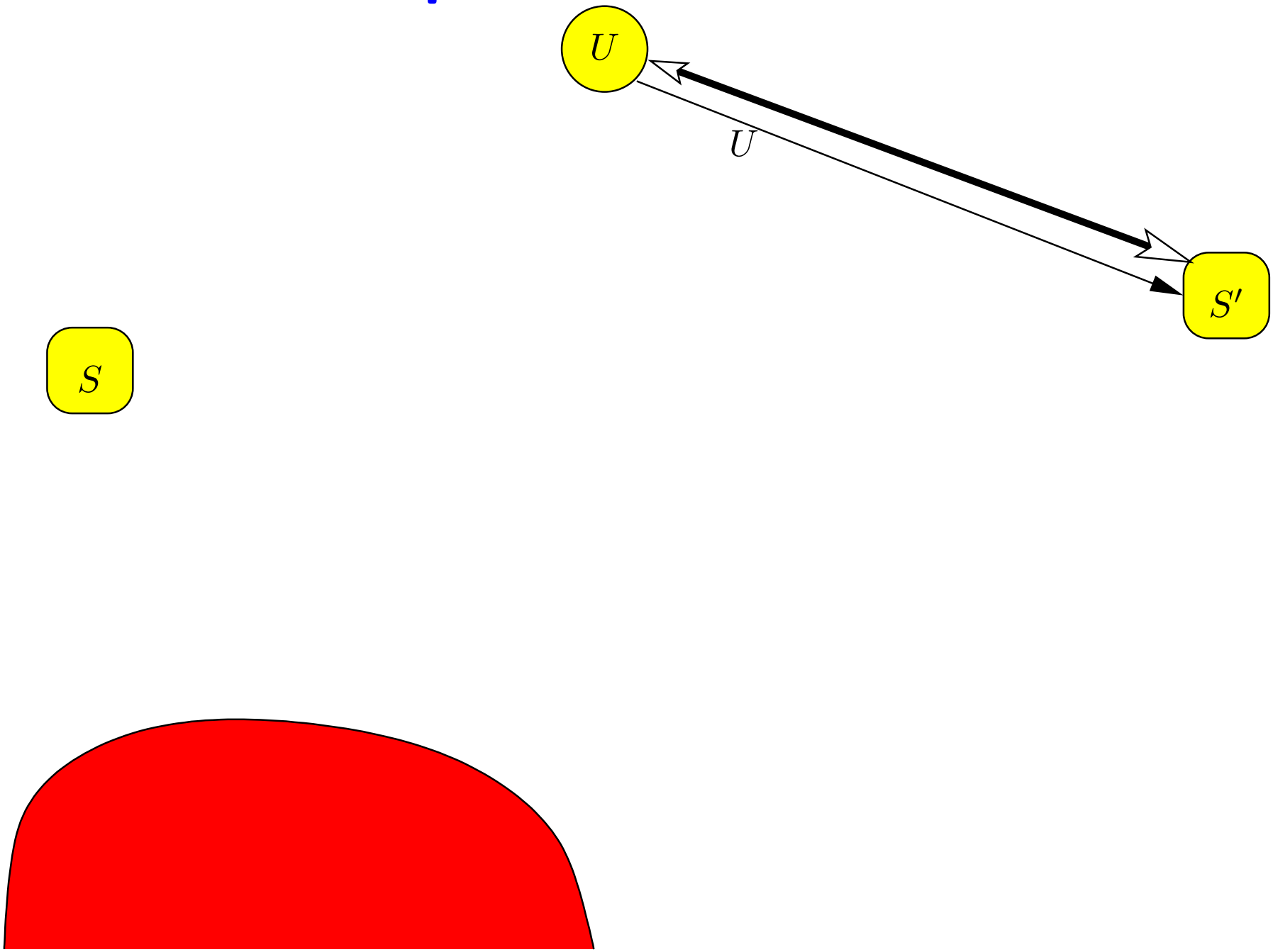
Relaxing the security model

- DigiDocService and Mobile Operator are just mediating parties.
- The security of the protocol should not depend on their honesty.

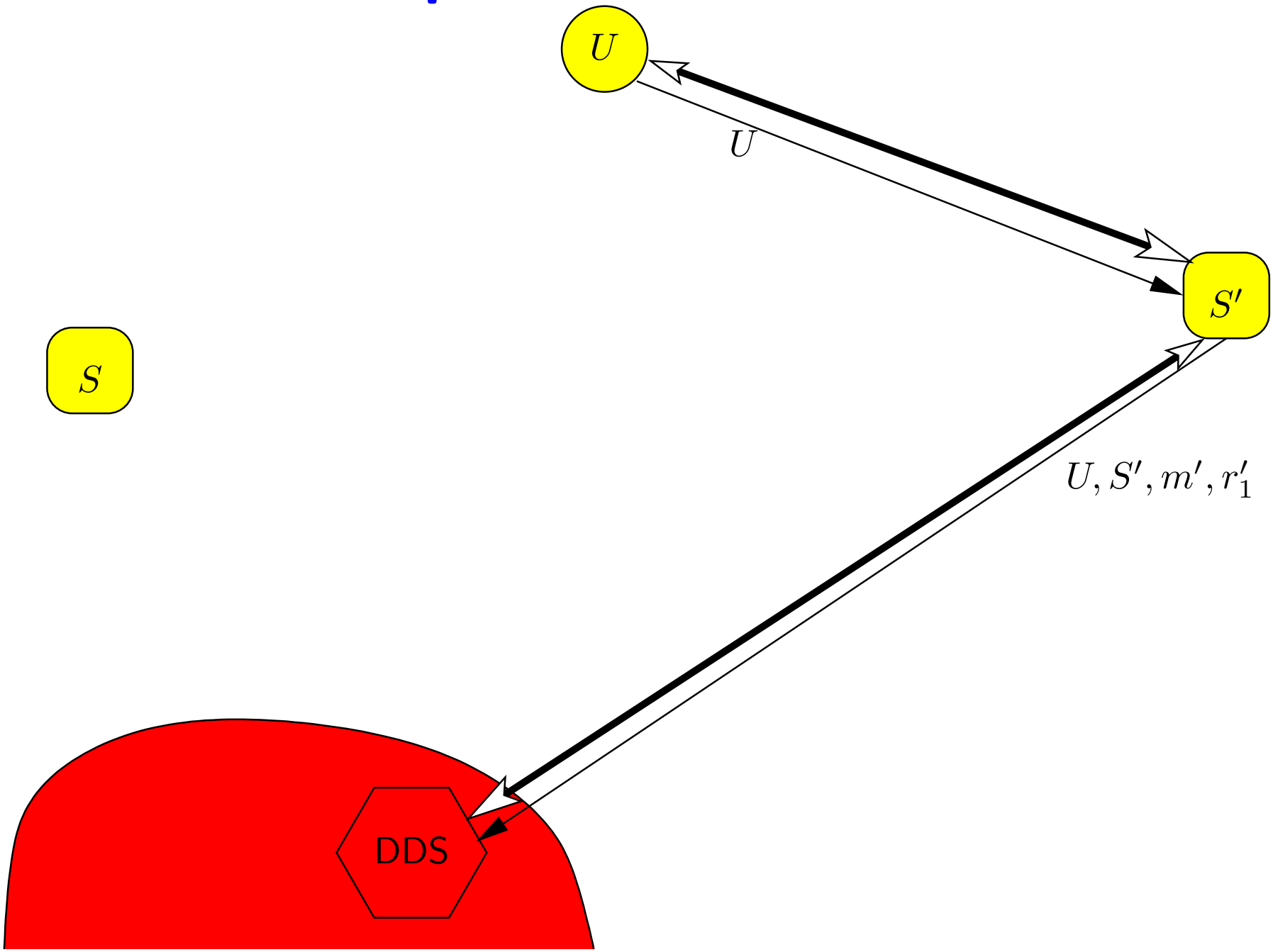
A possible scenario



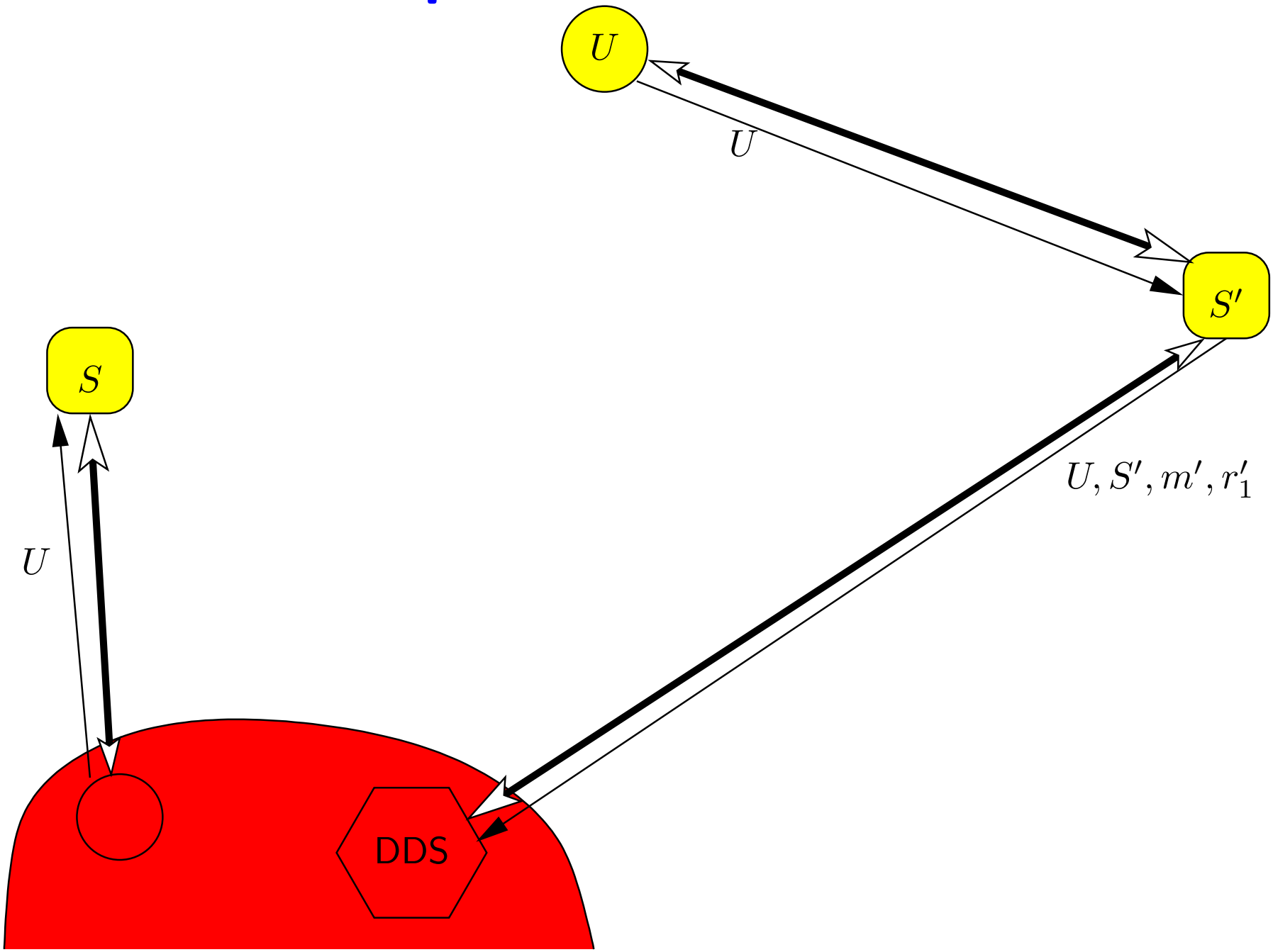
A possible scenario



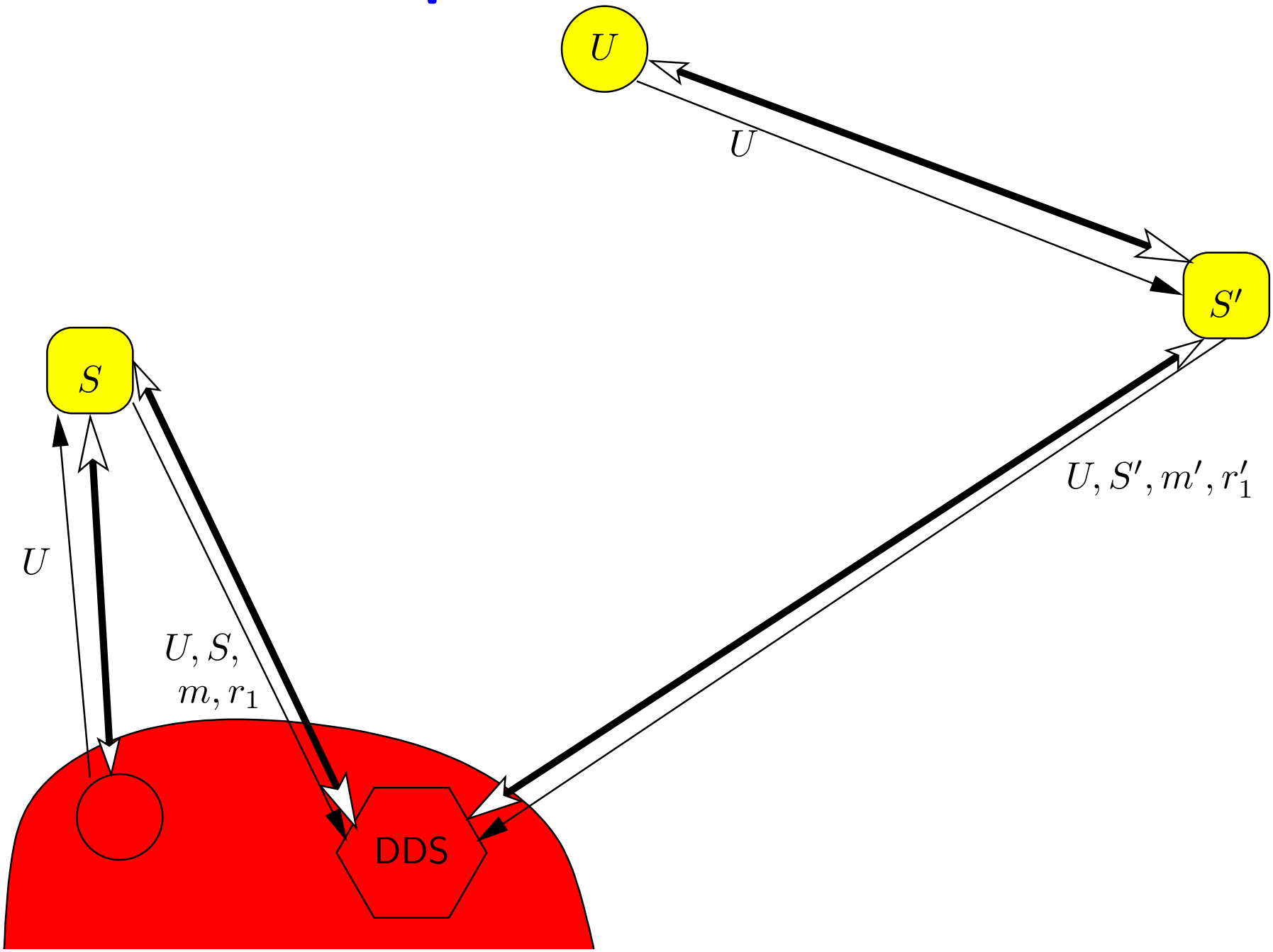
A possible scenario



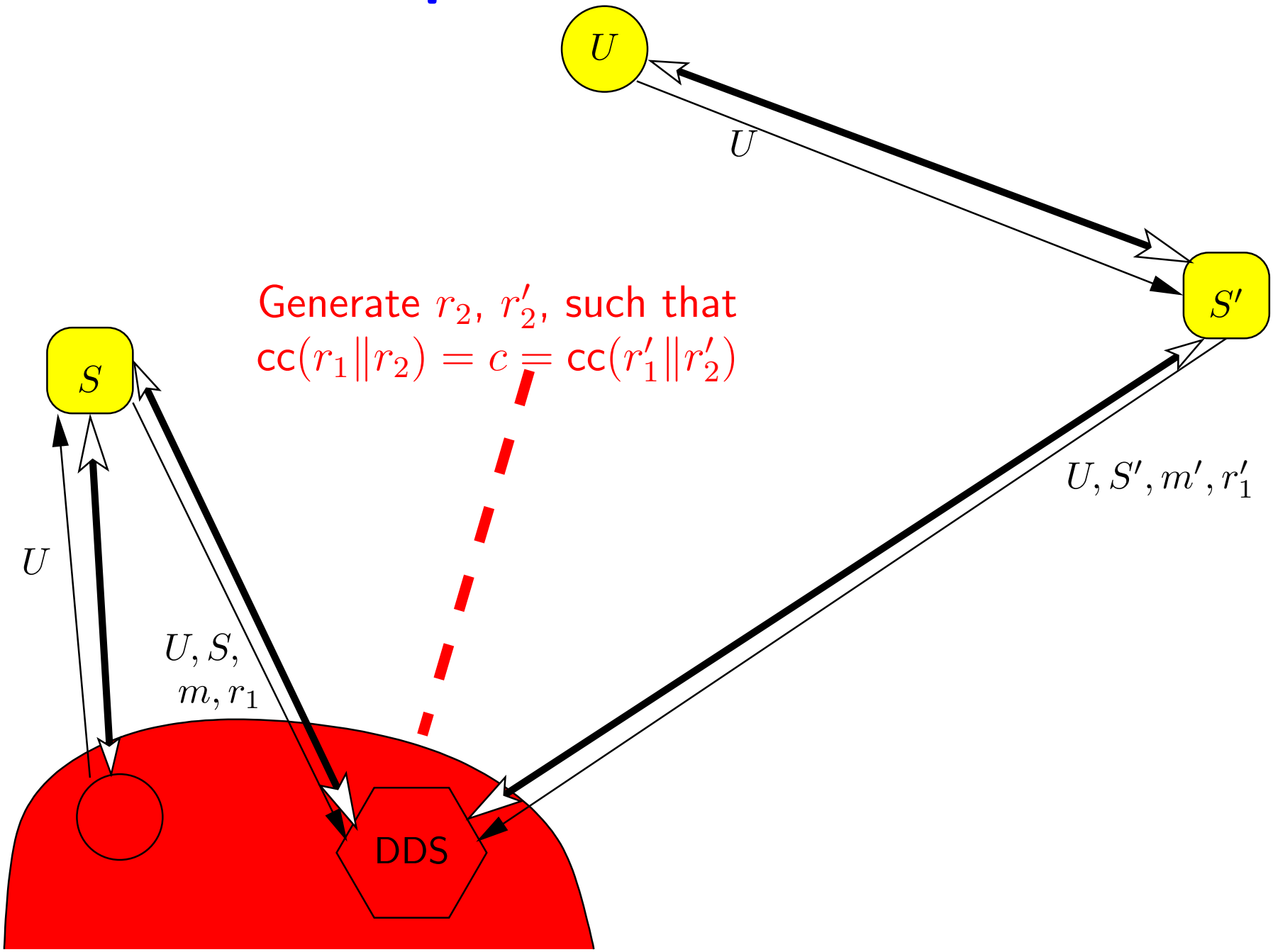
A possible scenario



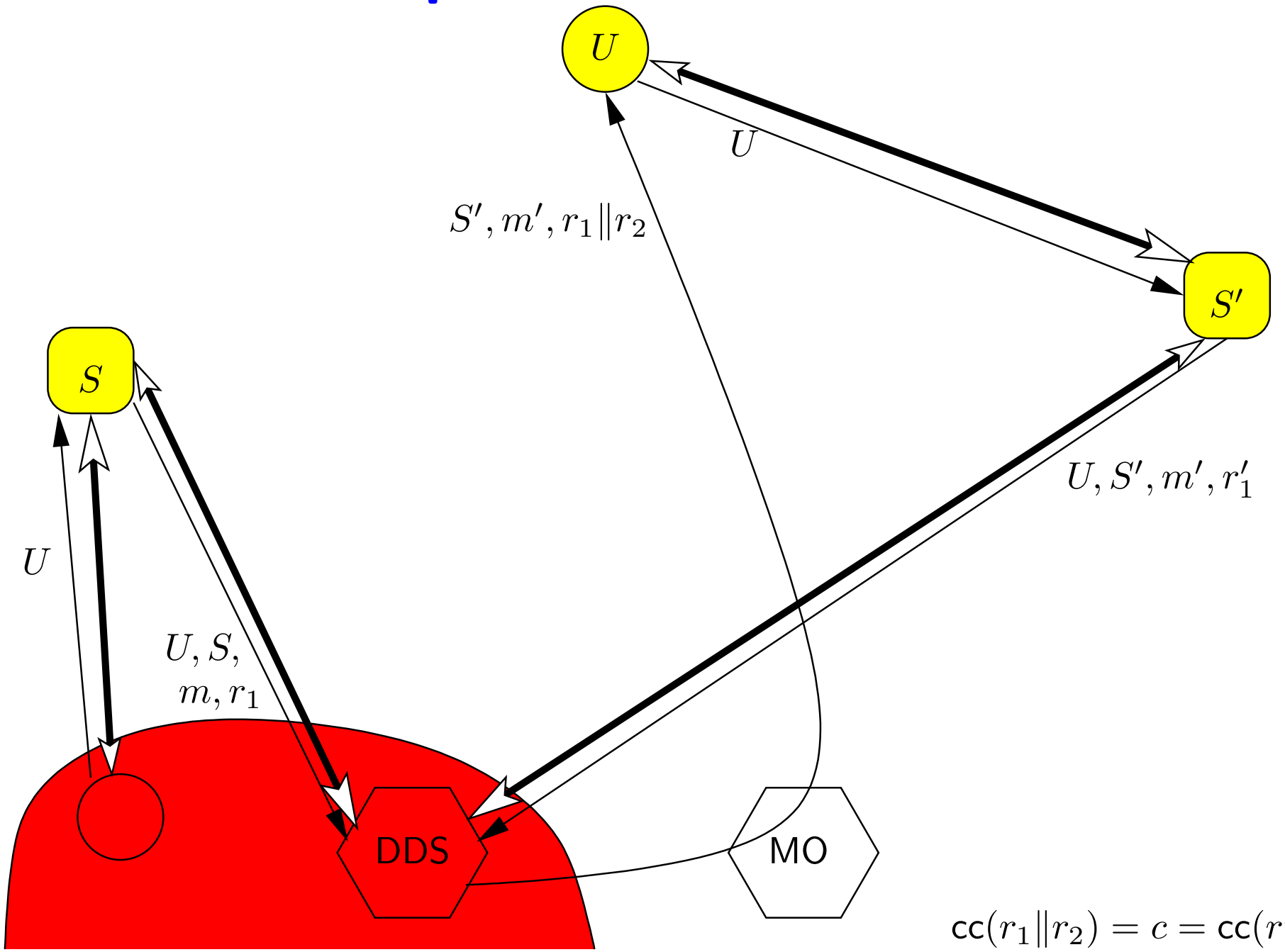
A possible scenario



A possible scenario

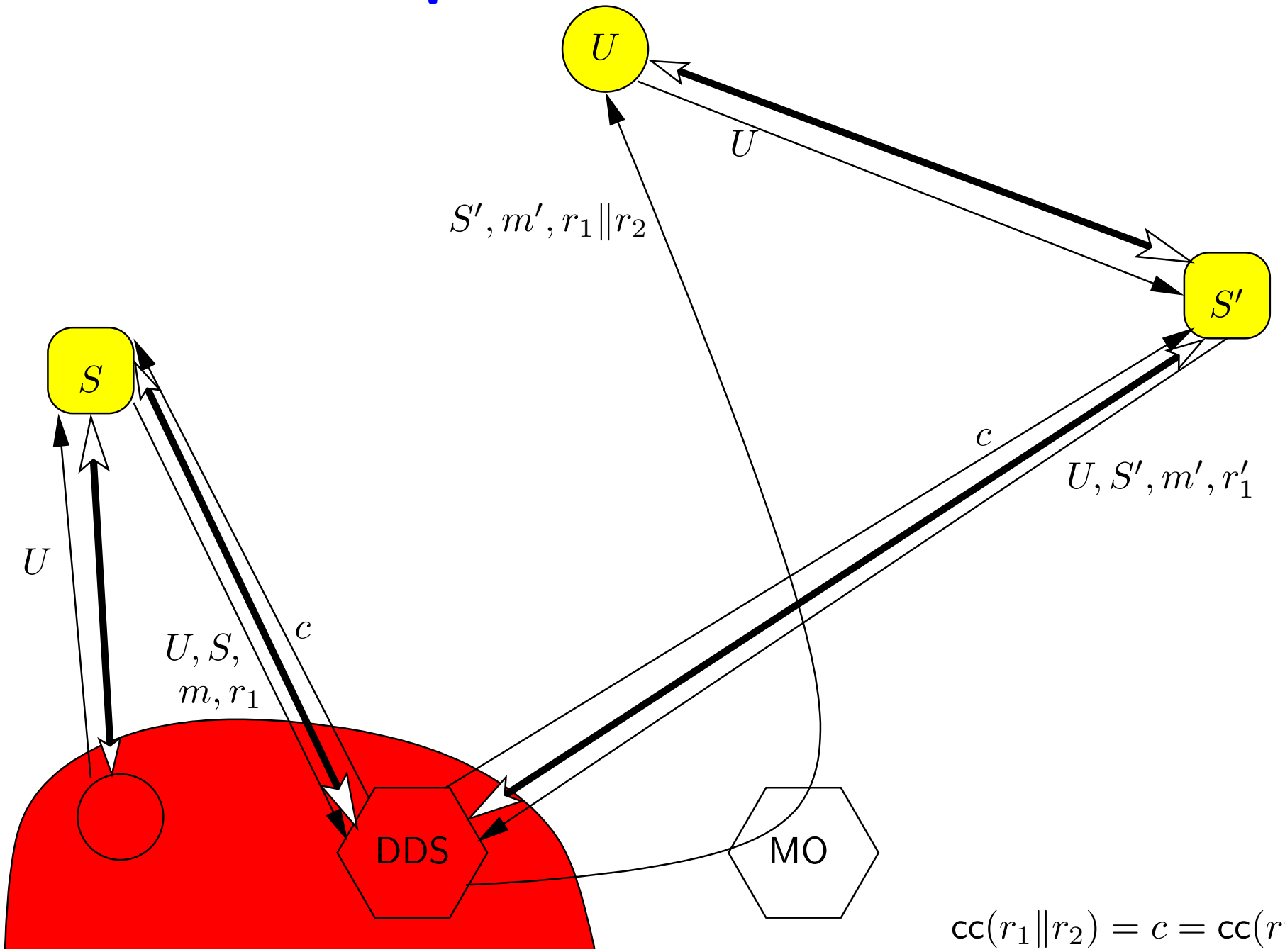


A possible scenario



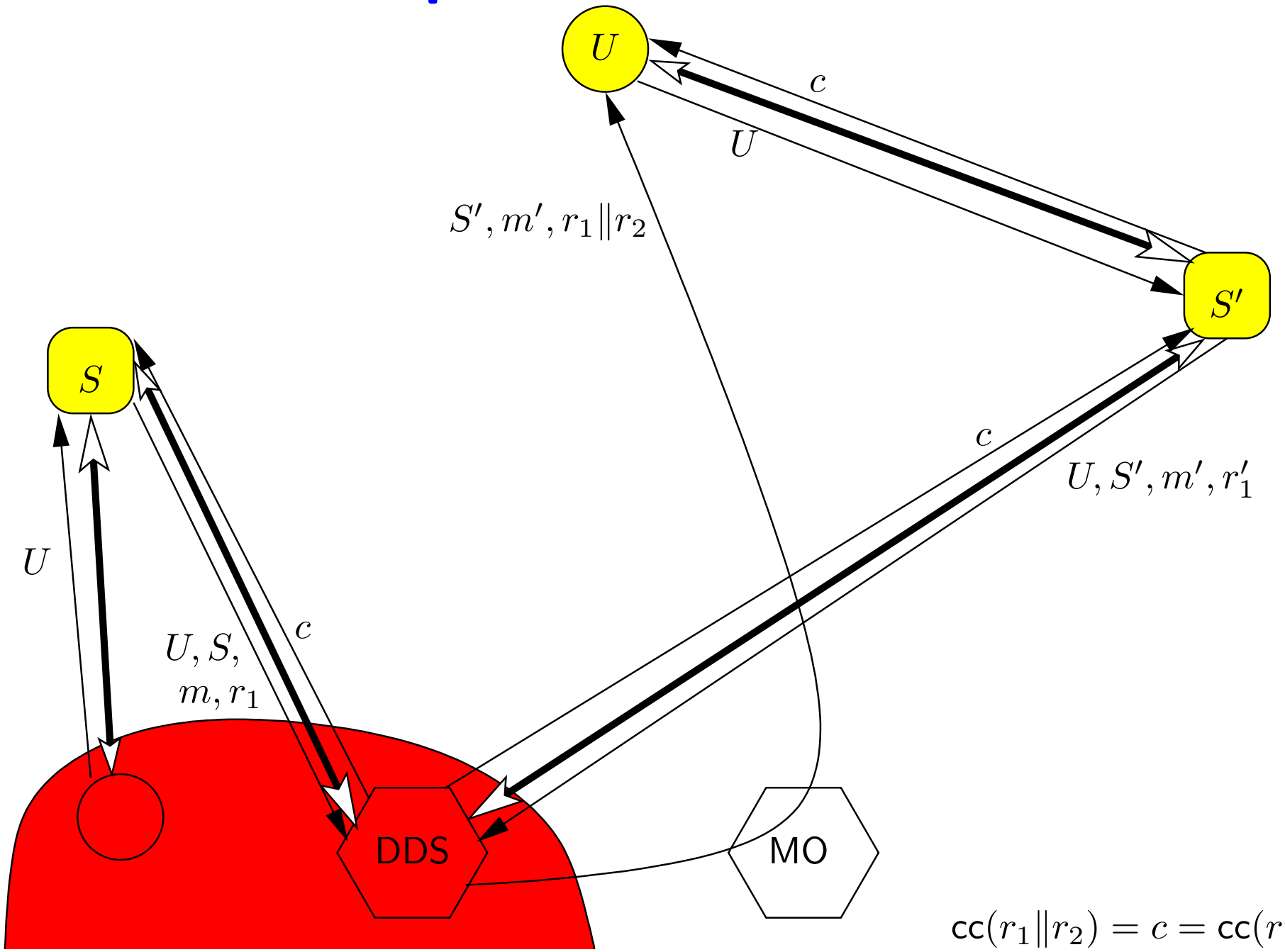
$\mathsf{cc}(r_1 \| r_2) = c = \mathsf{cc}(r_1' \| r_2')$

A possible scenario



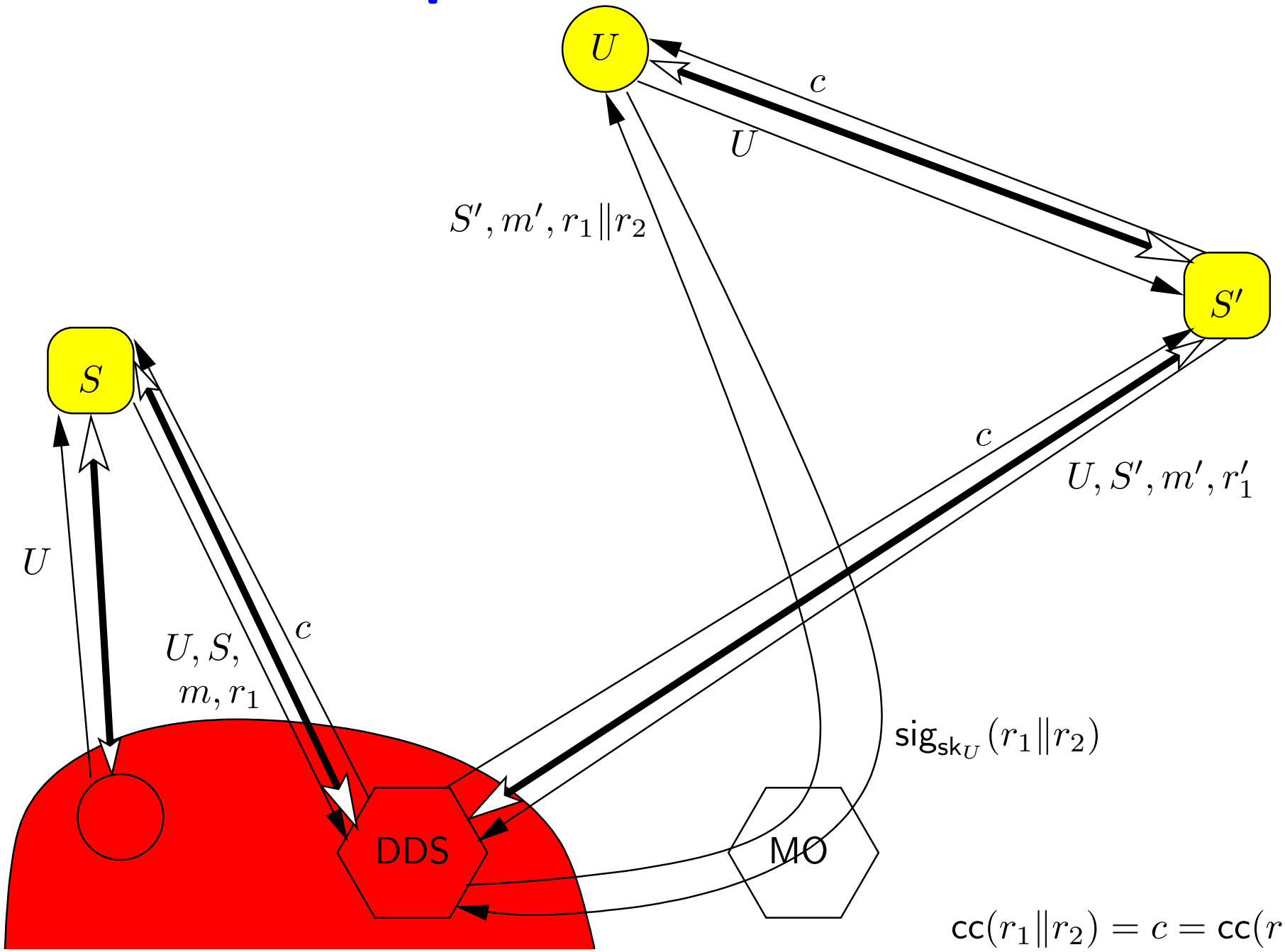
$\mathsf{cc}(r_1 \| r_2) = c = \mathsf{cc}(r_1' \| r_2')$

A possible scenario

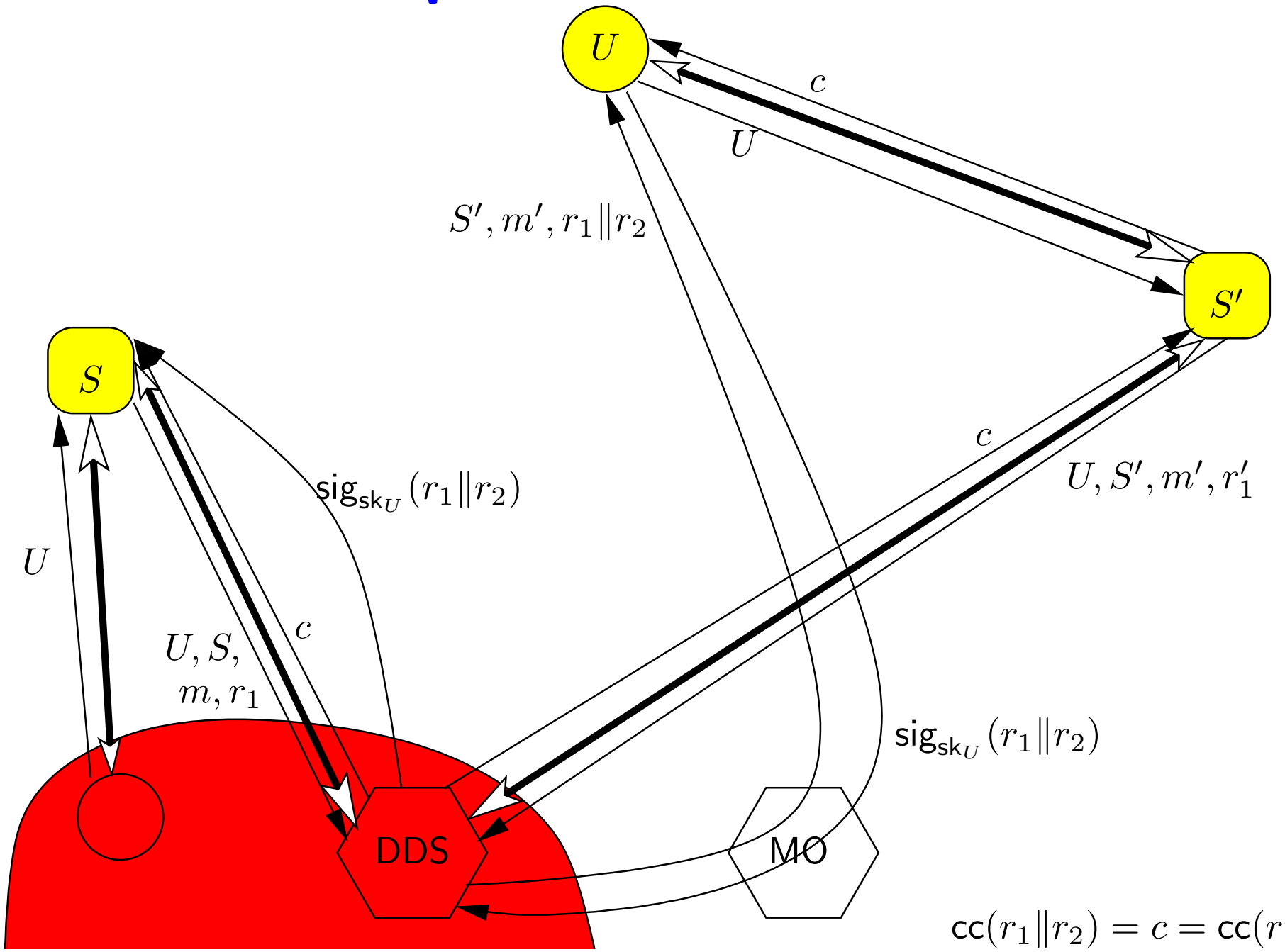


$\mathsf{cc}(r_1 \| r_2) = c = \mathsf{cc}(r_1' \| r_2')$

A possible scenario



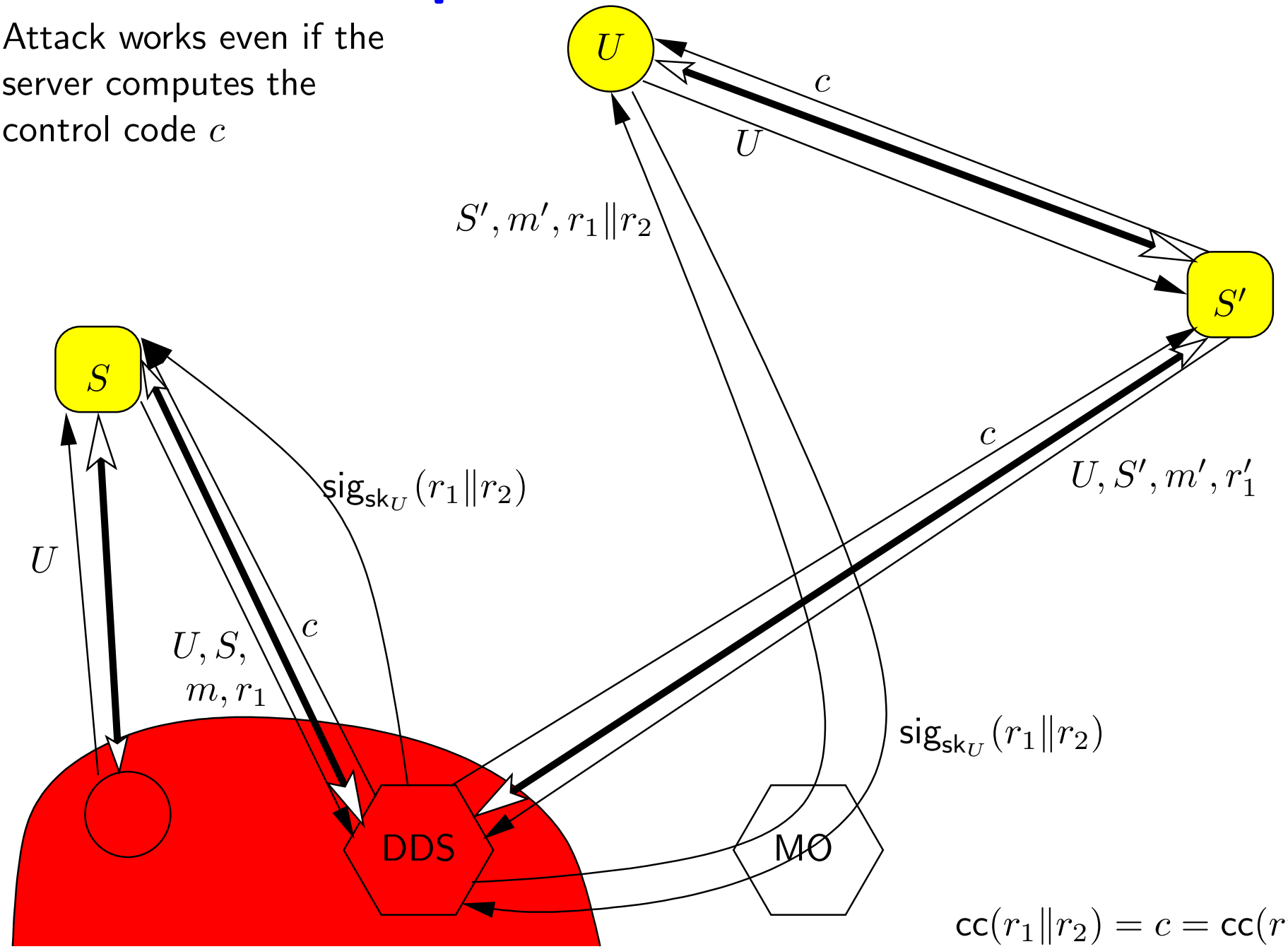
A possible scenario



$\mathsf{cc}(r_1 \| r_2) = c = \mathsf{cc}(r'_1 \| r'_2)$

A possible scenario

Attack works even if the server computes the control code c



$\mathsf{cc}(r_1\|r_2) = c = \mathsf{cc}(r'_1\|r'_2)$

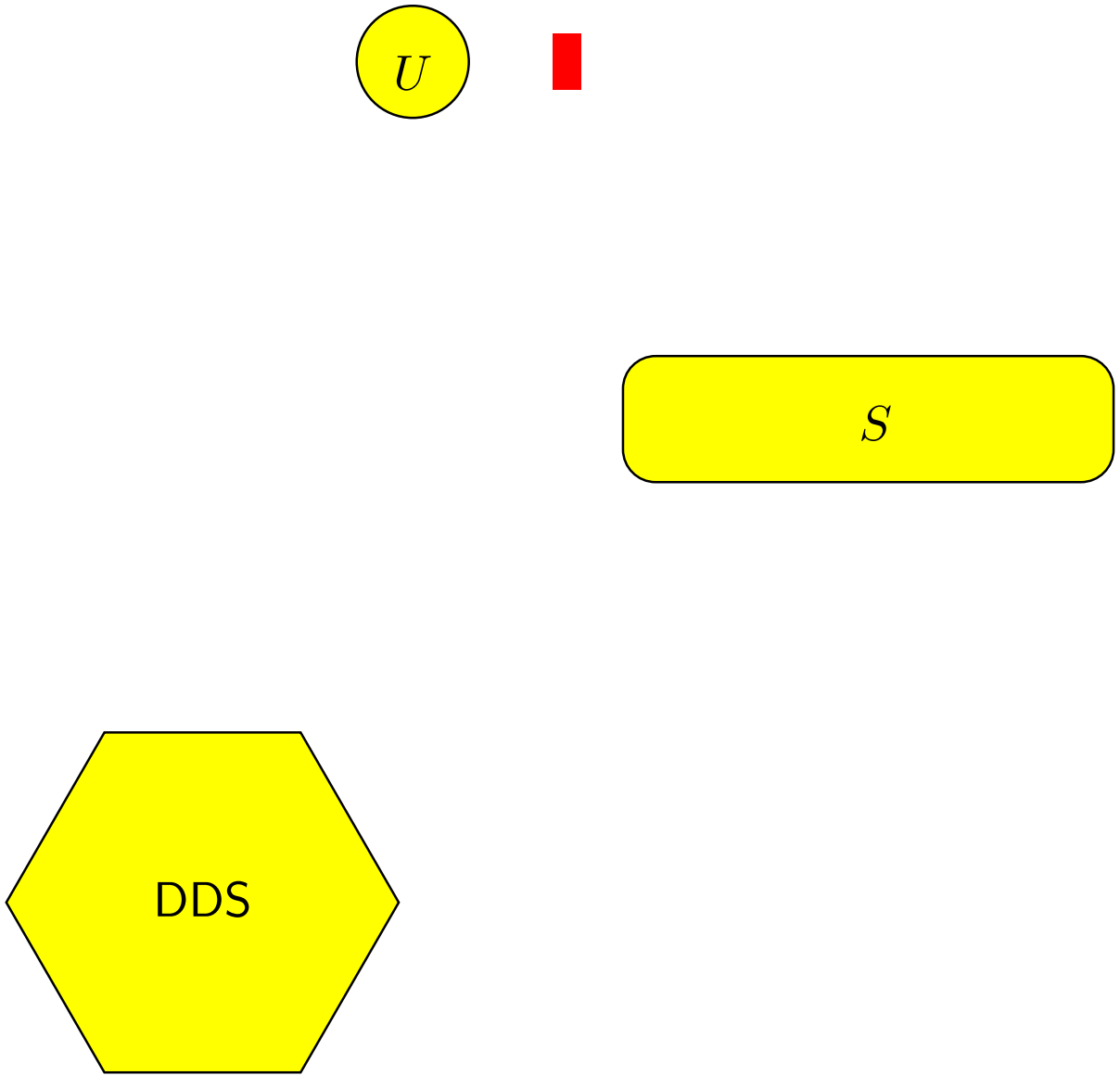
Malware in user's computer

- Full control over the client app. means knowing the TLS keys.
- Even a keylogger can cause a lot of harm if using the ID-card.

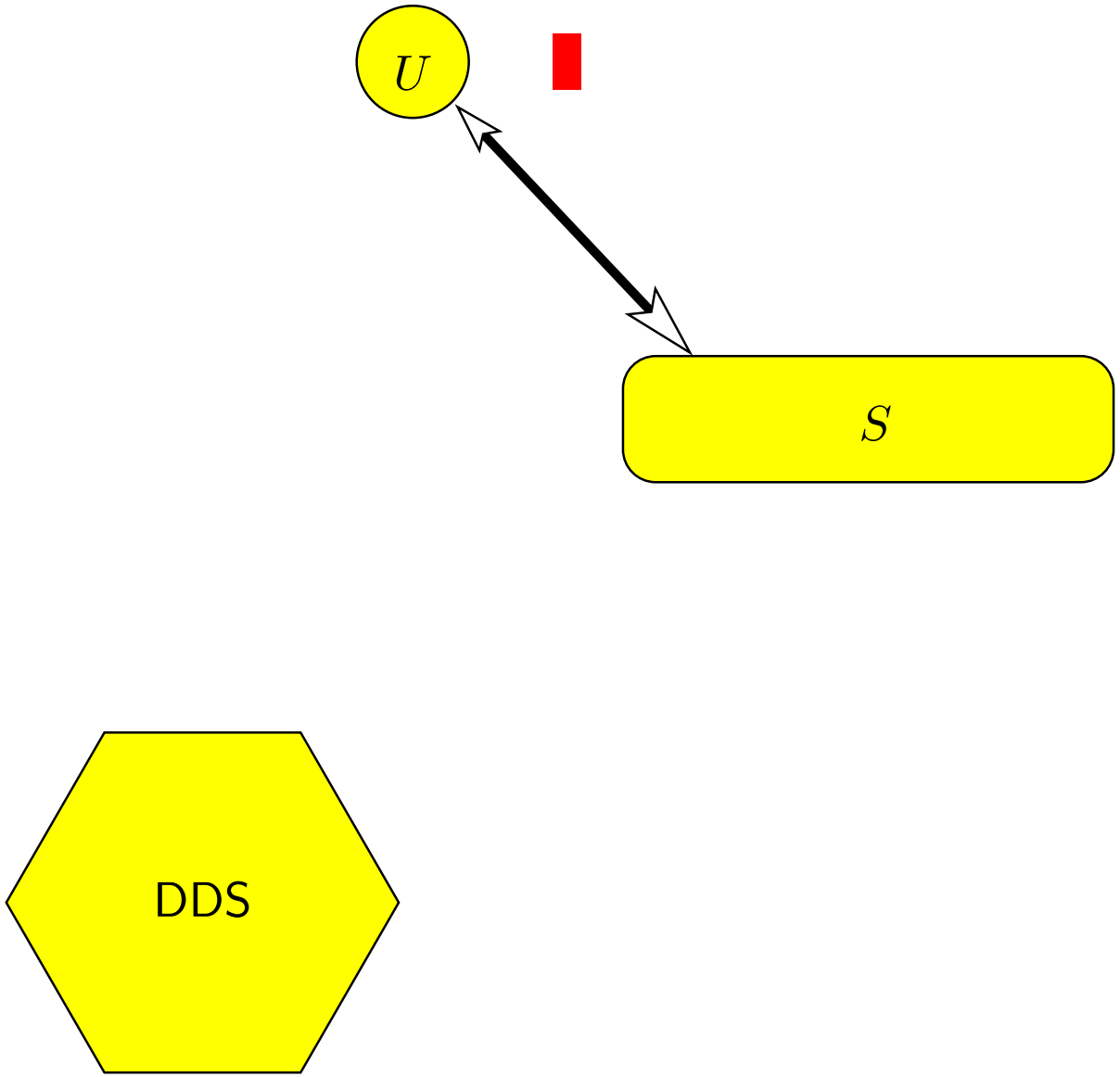
Malware in user's computer

- Full control over the client app. means knowing the TLS keys.
- Even a keylogger can cause a lot of harm if using the ID-card.
  - When using Mobile-ID, a keylogger in computer cannot record PINs.
- A similar level of control for the mobile-ID protocol might be the control over which control code is shown to the user.
- If the display manipulator also has network access then ...

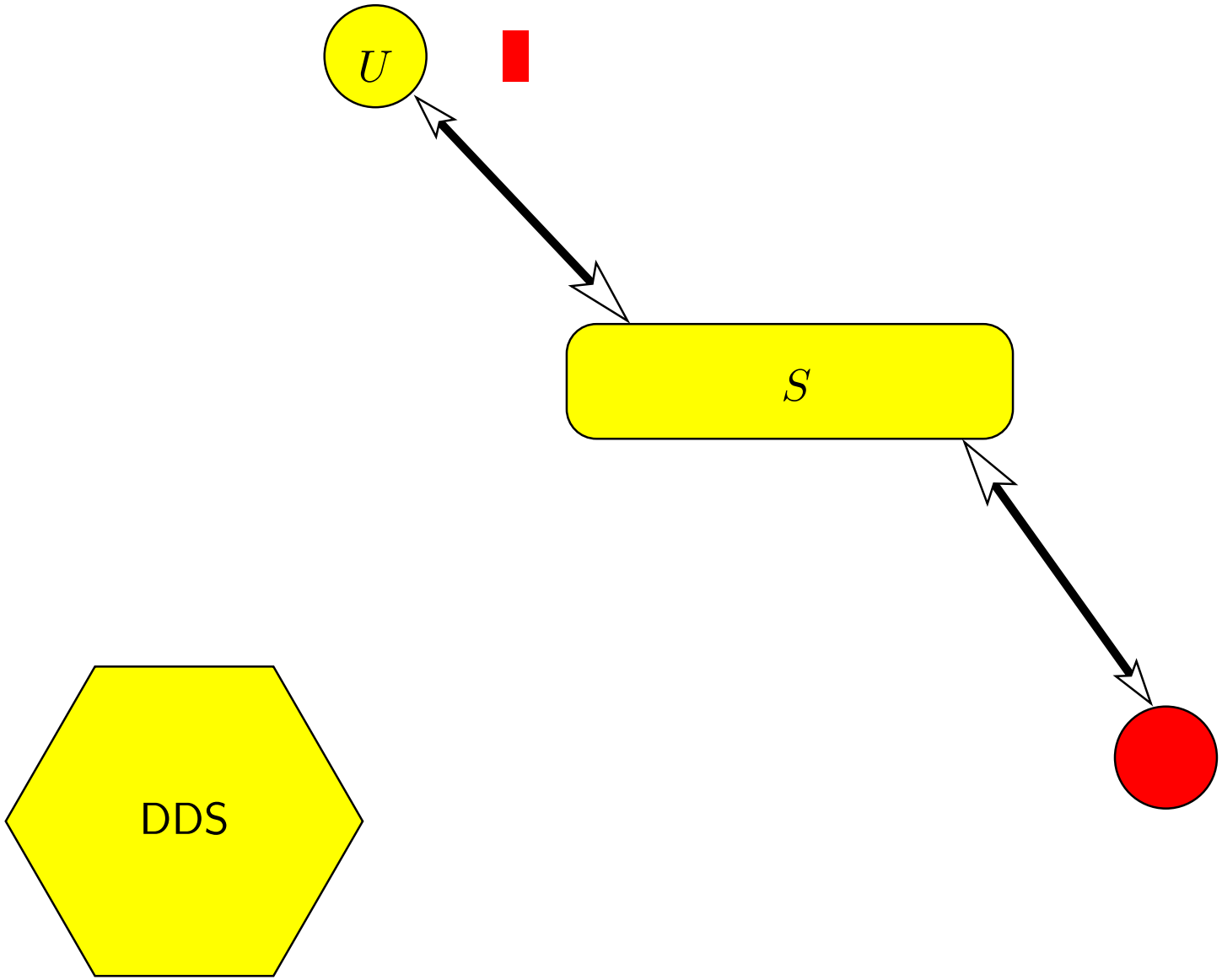
A possible scenario



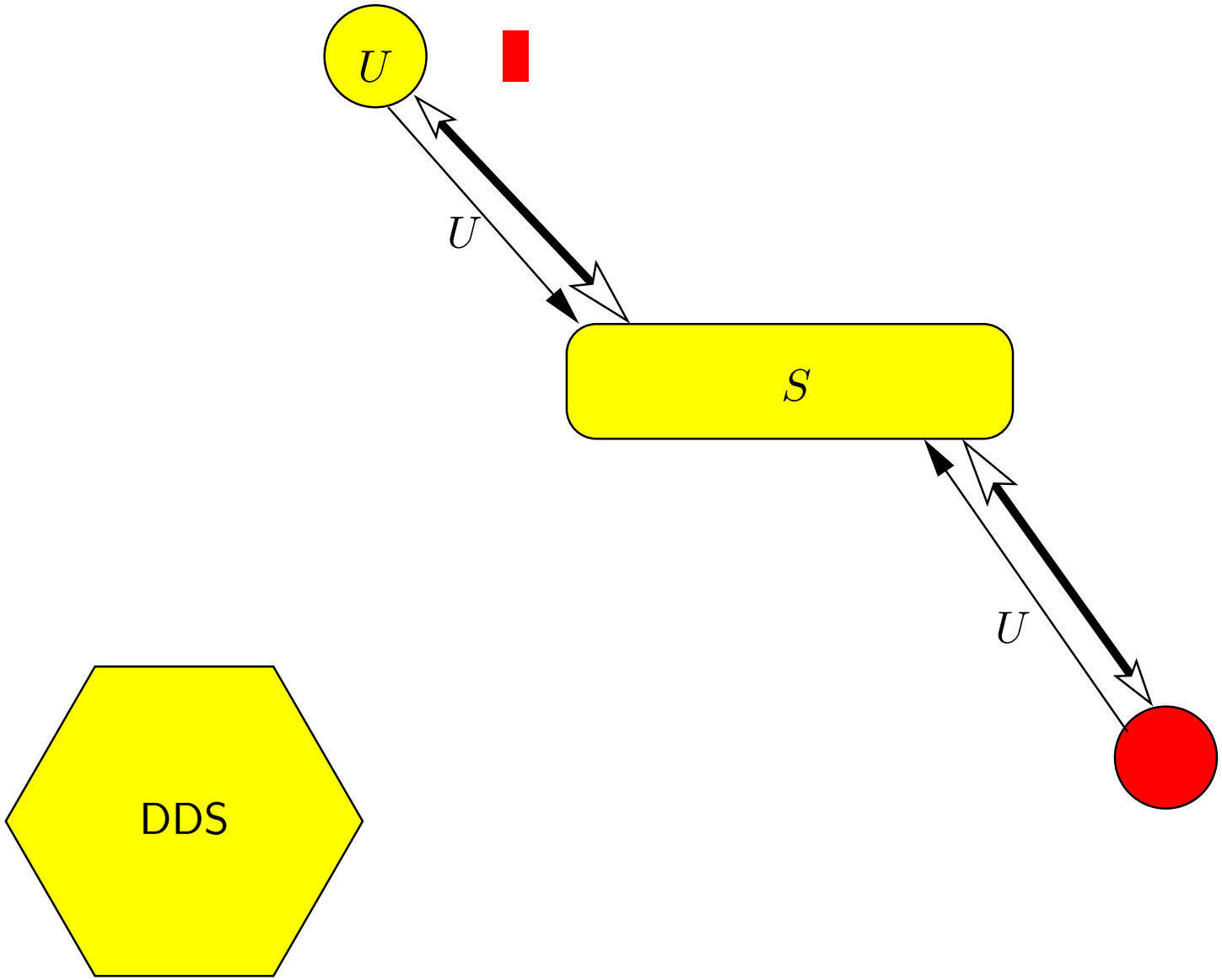
A possible scenario



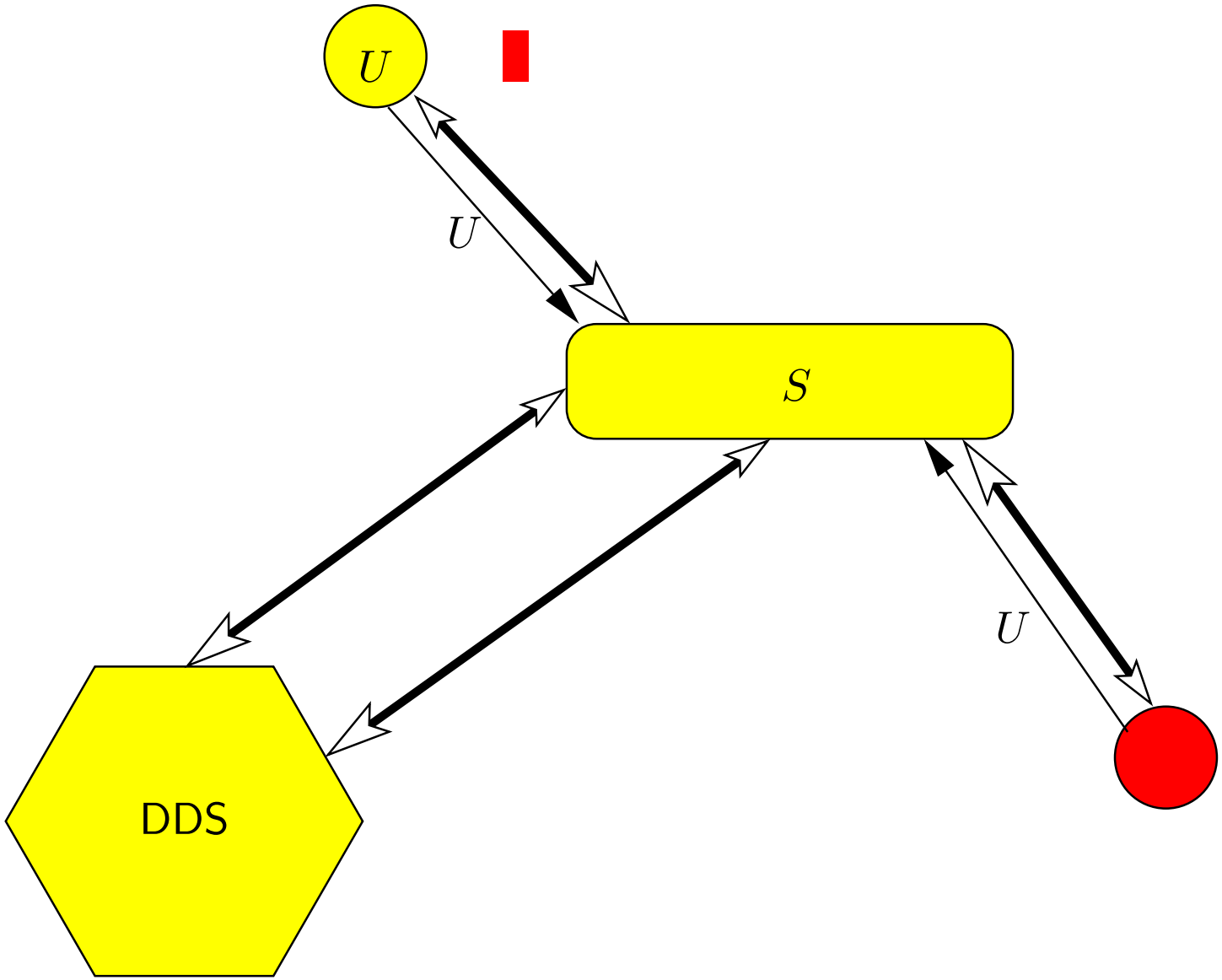
A possible scenario



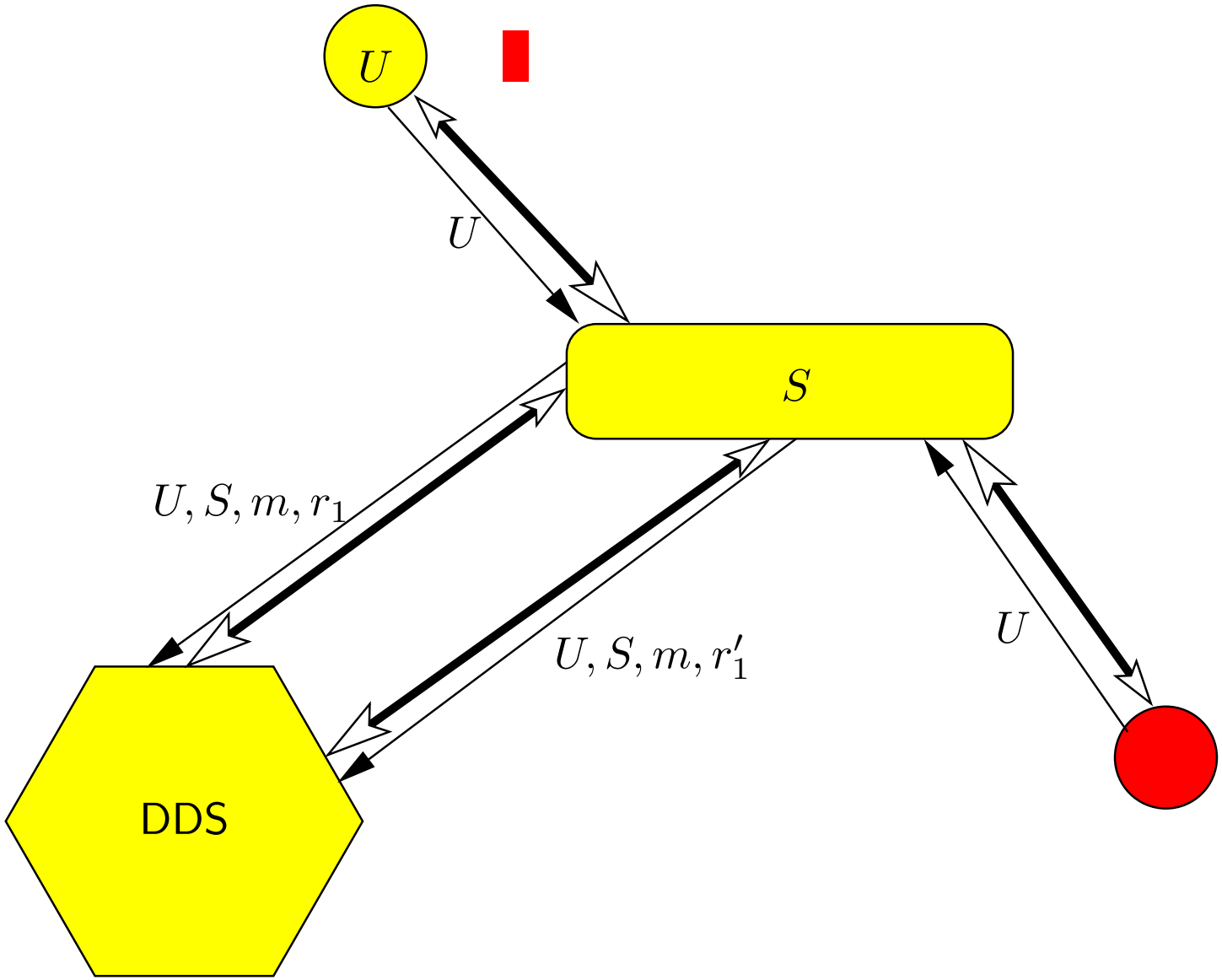
A possible scenario



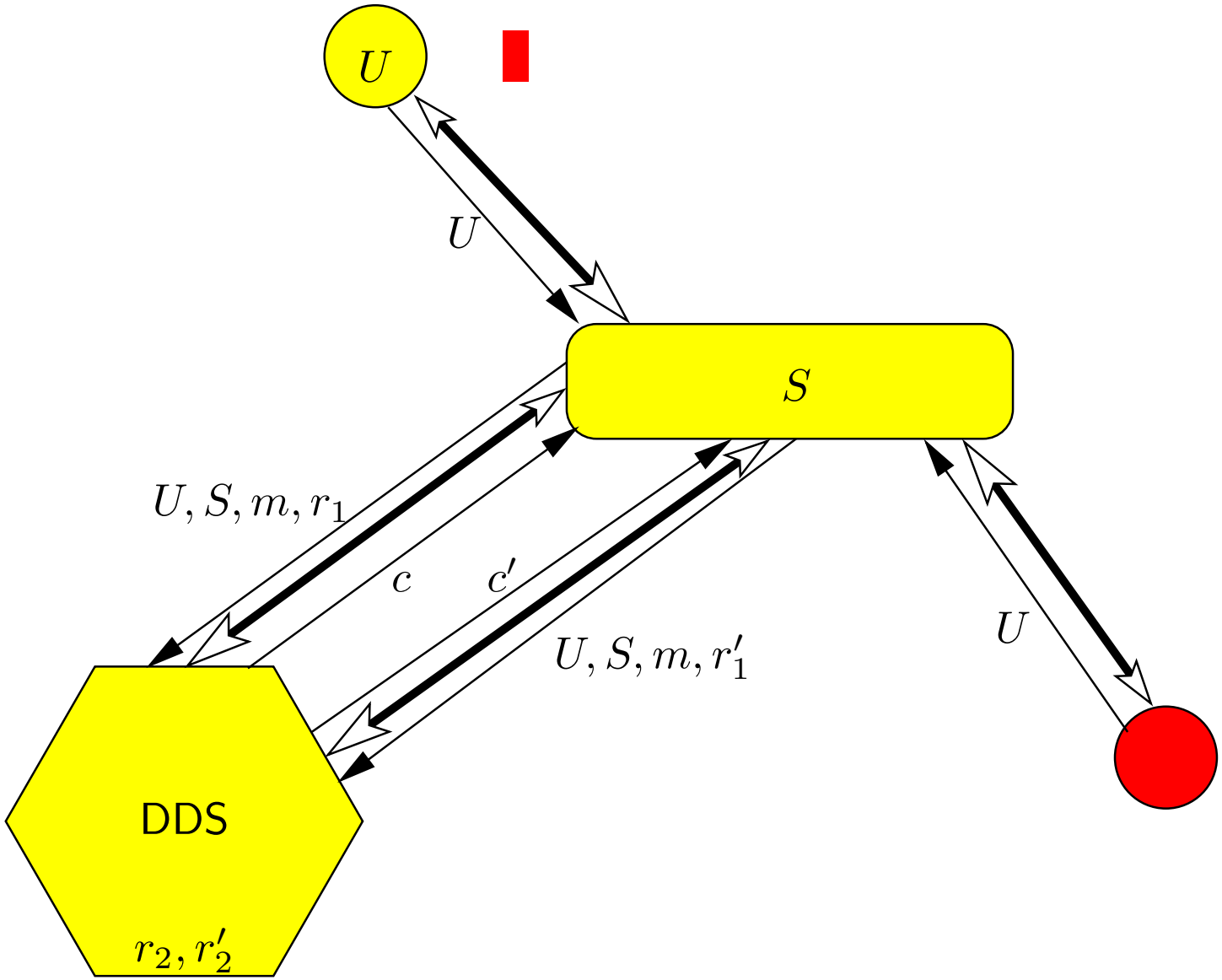
A possible scenario



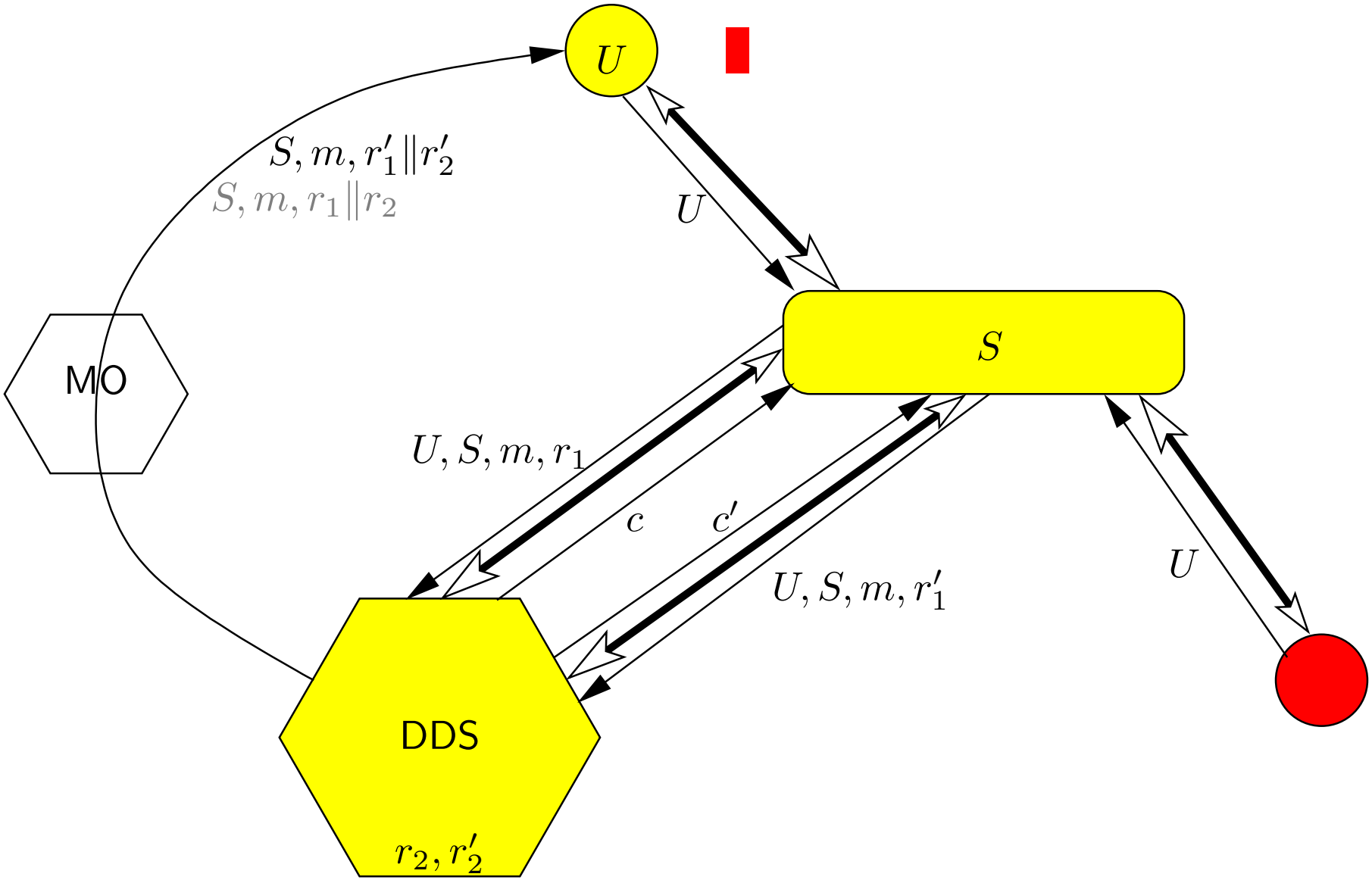
A possible scenario



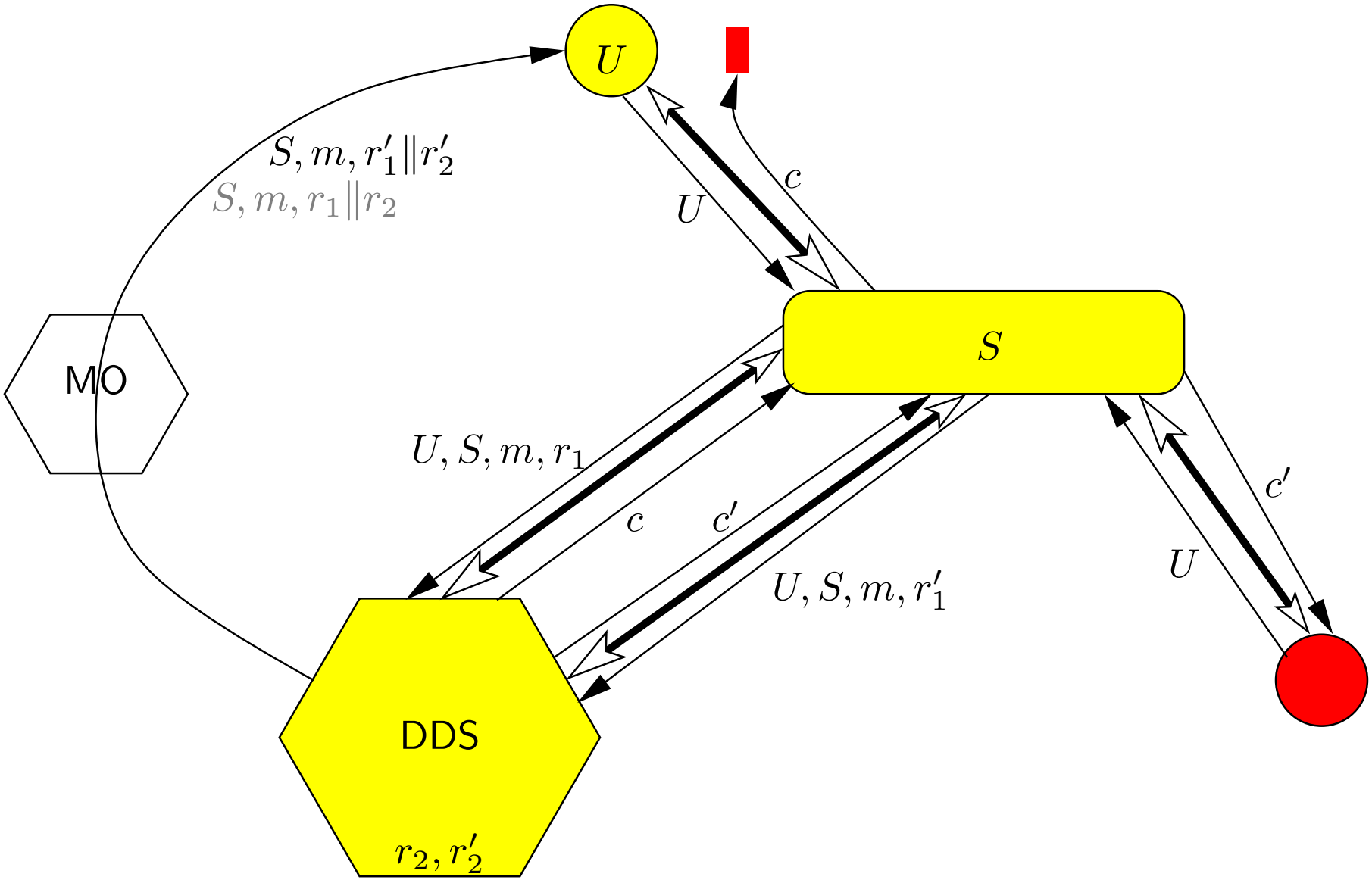
A possible scenario



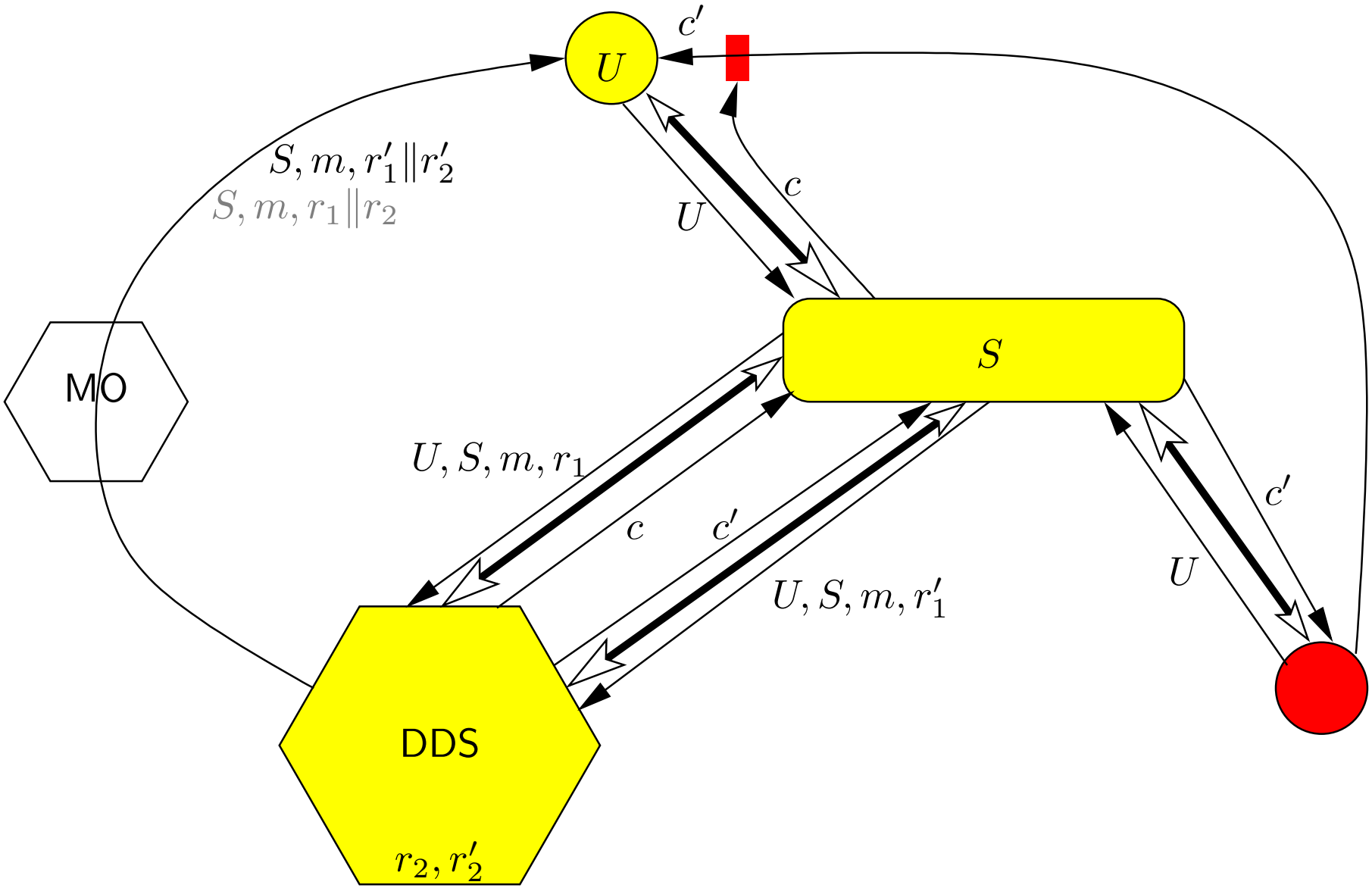
A possible scenario



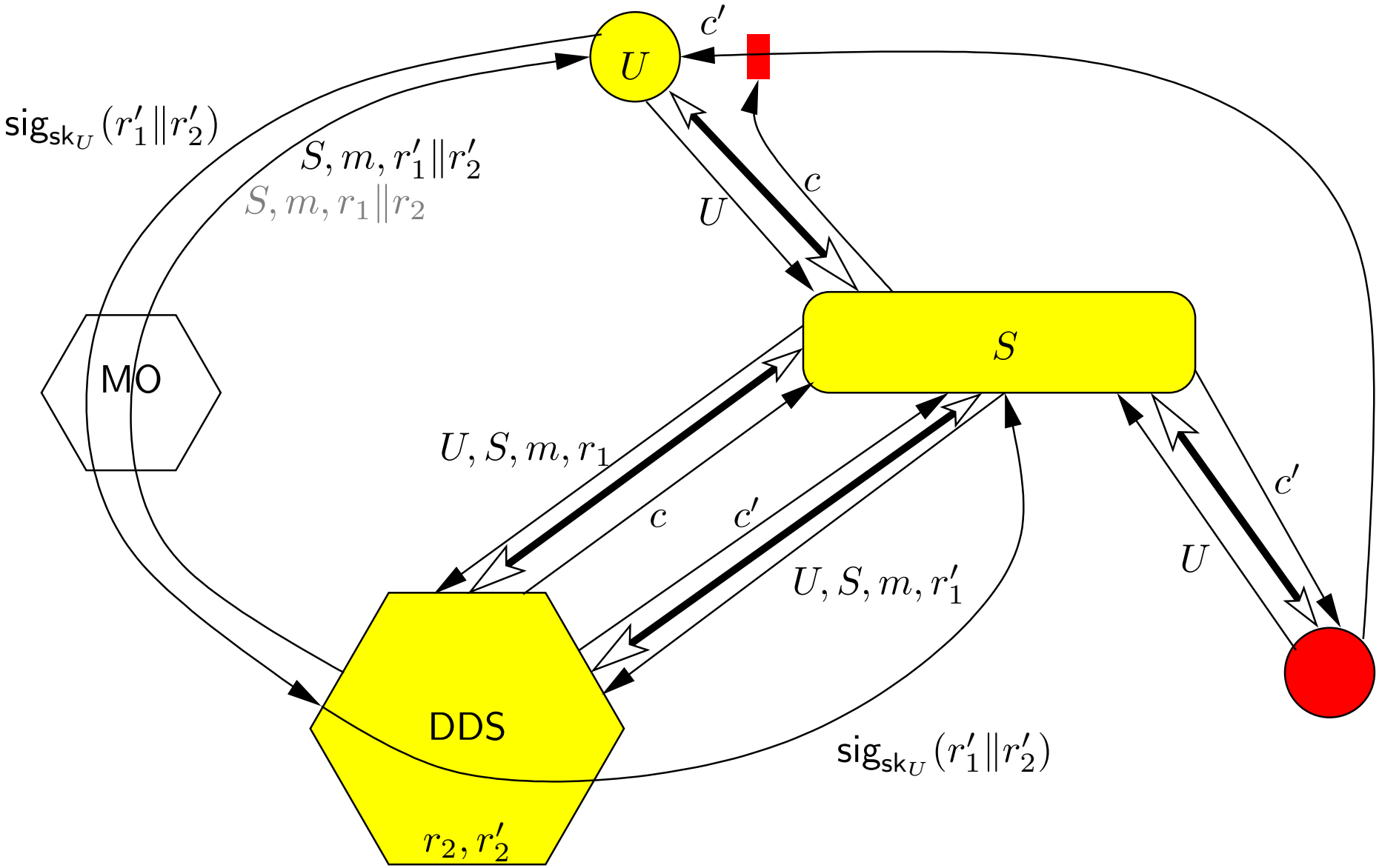
A possible scenario



A possible scenario



A possible scenario



Confusing the user about server identities

- If the user is duped to connect to a rogue site, then a man-in-the-middle attack is possible.
  - The attack gives the adversary access to the real site in the name of the user.
  - This attack is also present when authenticating with passwords (code cards, code calculators, one-time passwords, etc.)
  - This attack is not present when using the ID-card.

Issues with SIM-card software

- The SIM-card software shows embedded newlines in m as line breaks.
  - The server can construct a message m that obscures the actual control code.
  - Not exploitable if the DigiDocService is honest; but must be considered otherwise.

Suggested changes

- Instead of signing the challenge r, sign (r, S).
- Whole challenge r should be chosen and the control code CC_1 computed by S.
  - S must avoid control code collsions in parallel sessions with the same U.
- Change the way m and CC_2 are shown on the phone screen and/or educate users such that CC_2 will not be obscured.

Still no protection against trojans in phone or computer.