Direct Anonymous Attestation (DAA): Ensuring privacy with corrupt administrators

Ben Smyth Mark Ryan Liqun Chen

School of Computer Science,
University of Birmingham, UK
{B.A.Smyth,M.D.Ryan}@cs.bham.ac.uk

HP Laboratories,
Bristol, UK
liqun.chen@hp.com

Tuesday, 2nd July 2007

Trusted Computing

- Cryptographic guarantees about the state of a remote platform
- Allowing informed decisions as to whether to trust the platform
- Requires hardware device called a Trusted Platform Module (TPM)
- Example applications:
  - Mobile *ad hoc* networks
  - Grid computing
  - Corporate digital rights management

Treacherous computing?

- Privacy advocates (and TCG members) voiced concerns
- Solution: server learns that a platform is trusted and not which particular one

Privacy

- **Anonymity:** state of not being identifiable within a set of agents
- **Unlinkability:** inability to link actions performed by the same agent

Direct Anonymous Attestation (DAA)

Developed in collaboration between Ernie Brickell (Intel), Jan Camenisch (IBM) & Liqun Chen (HP)

TPM specification history

v1.1: Trusted third party
v1.2: Direct Anonymous Attestation (DAA)

What's in a name?

Direct proof: without trusted third party
Anonymous: platform/user identity not disclosed
Attestation: membership claim (from TPM)

DAA high level overview

Join protocol

- TPM chooses a secret message f
- Obtain signature (aka attestation) on f from the *issuer*

Sign/verify protocol

- Demonstrate knowledge of attestation to a *verifier*

Terminology

TPM: A hardware device which is trustworthy
Host/platform: A device with an embedded TPM
Issuer: Responsible for adding group members
Verifier: An entity which remotely authenticates a host

DAA security properties

1. Only a trusted platform is able to authenticate
2. Privacy of a non-corrupt host is guaranteed by the sign/verify protocol:
   - 2.1 Interactions are anonymous
   - 2.2 Linkability (of transactions) is controlled by the user
3. Privacy is restored to a corrupted host if malicious software is removed
4. Mechanism to detect rogue members

Provably secure

DAA has been shown to be secure in the provable security model

Inadequacy of provable security

Provable security aims to show the difficulty of breaking a scheme is as difficult as solving a supposedly difficult problem (e.g. integer factorisation)

Provable security on its own is insufficient:

- Effective attacks to not solve difficult problems, they discover weaknesses in the protocol

Notation

- Encryption

$$\{msg\}_K$$

- Zero knowledge proofs of knowledge

$$PK\{(\alpha, \beta, \gamma) : y = g^{\alpha} h^{\beta} \wedge \tilde{y} = \tilde{g}^{\alpha} \tilde{h}^{\gamma} \wedge \alpha \in [u, v]\}$$

- Signature proofs of knowledge

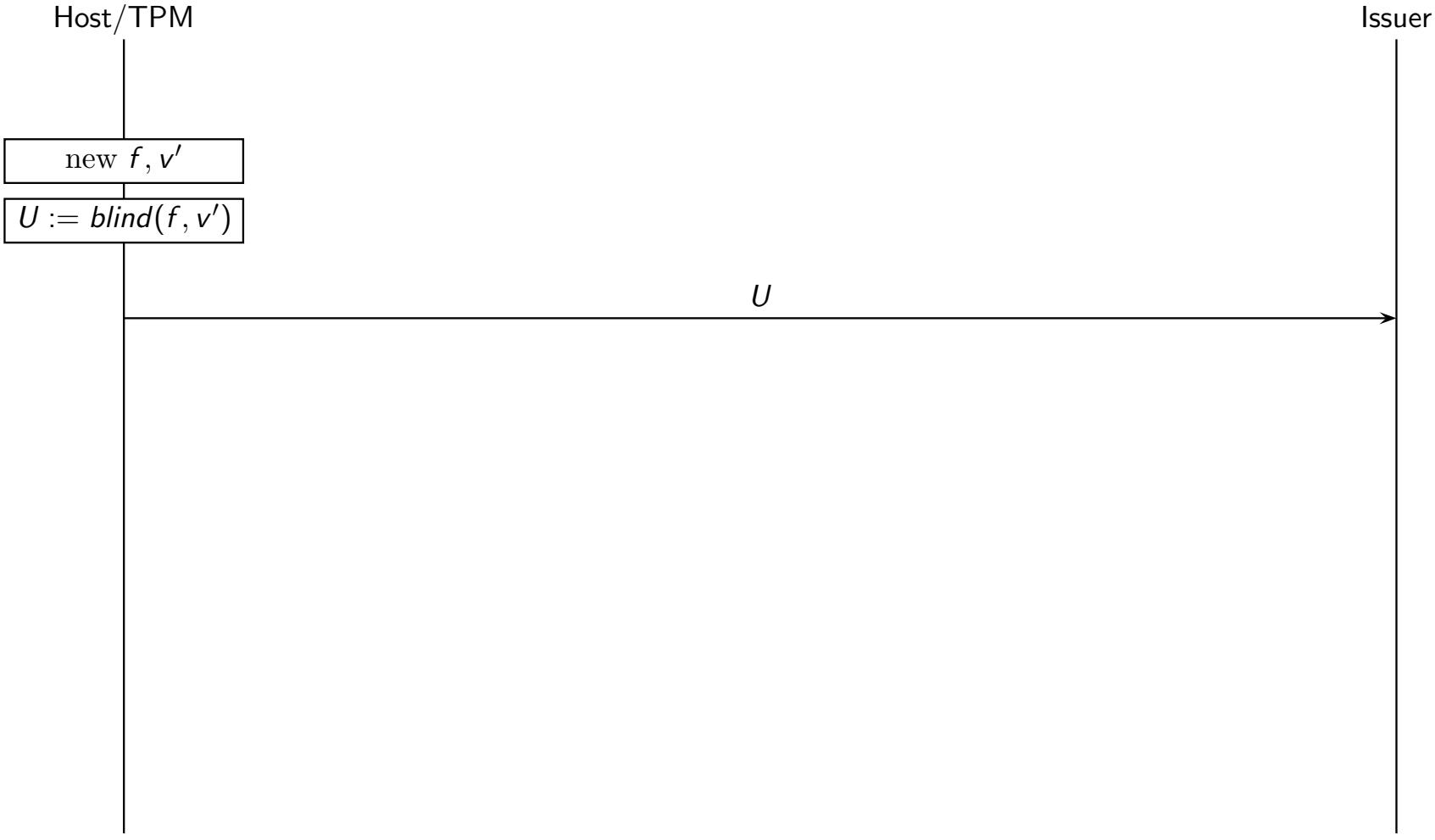
$$SPK\{(\alpha, \beta, \gamma) : y = g^{\alpha} h^{\beta} \wedge \tilde{y} = \tilde{g}^{\alpha} \tilde{h}^{\gamma}\}(m)$$

Join Protocol

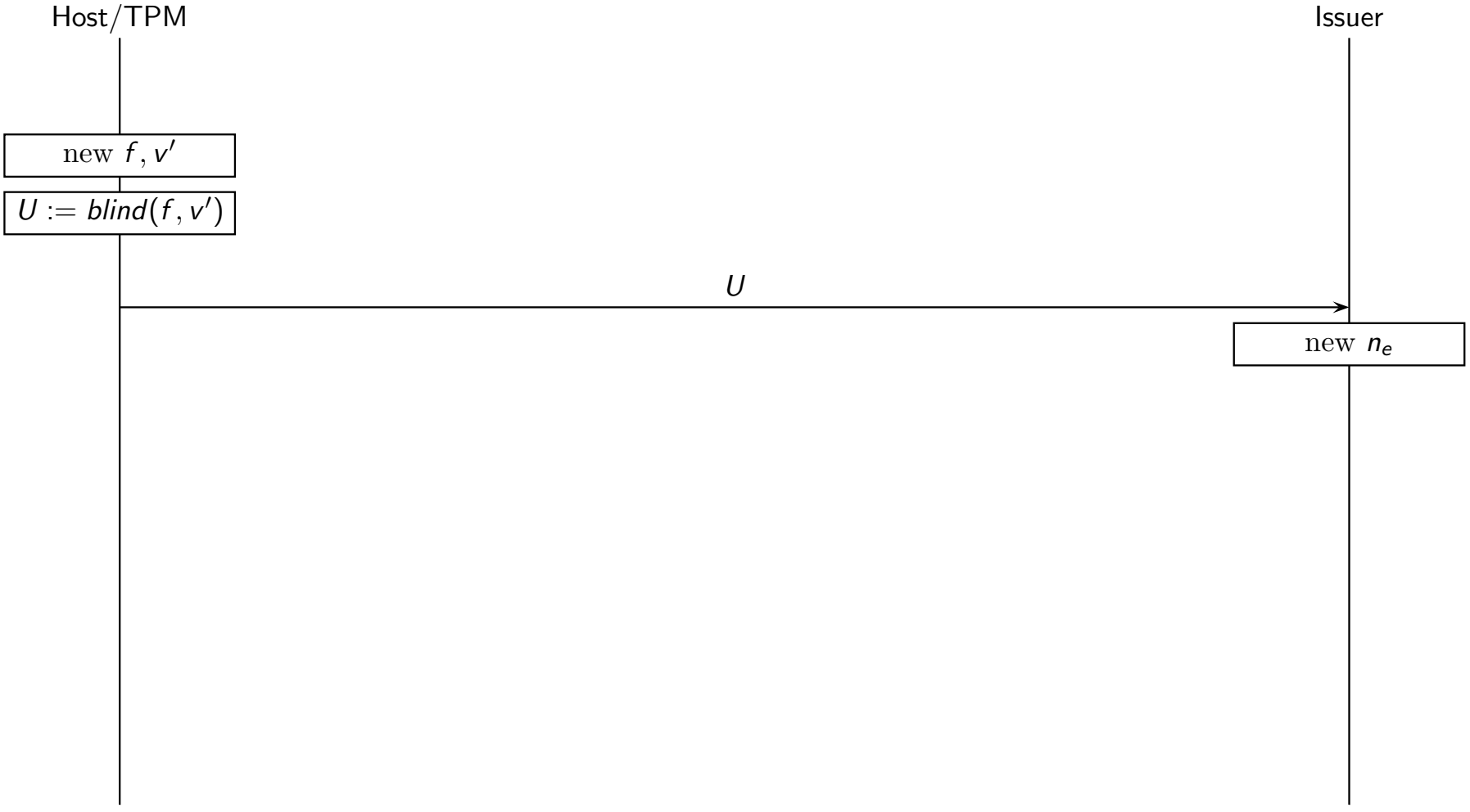
Join Protocol

Join Protocol

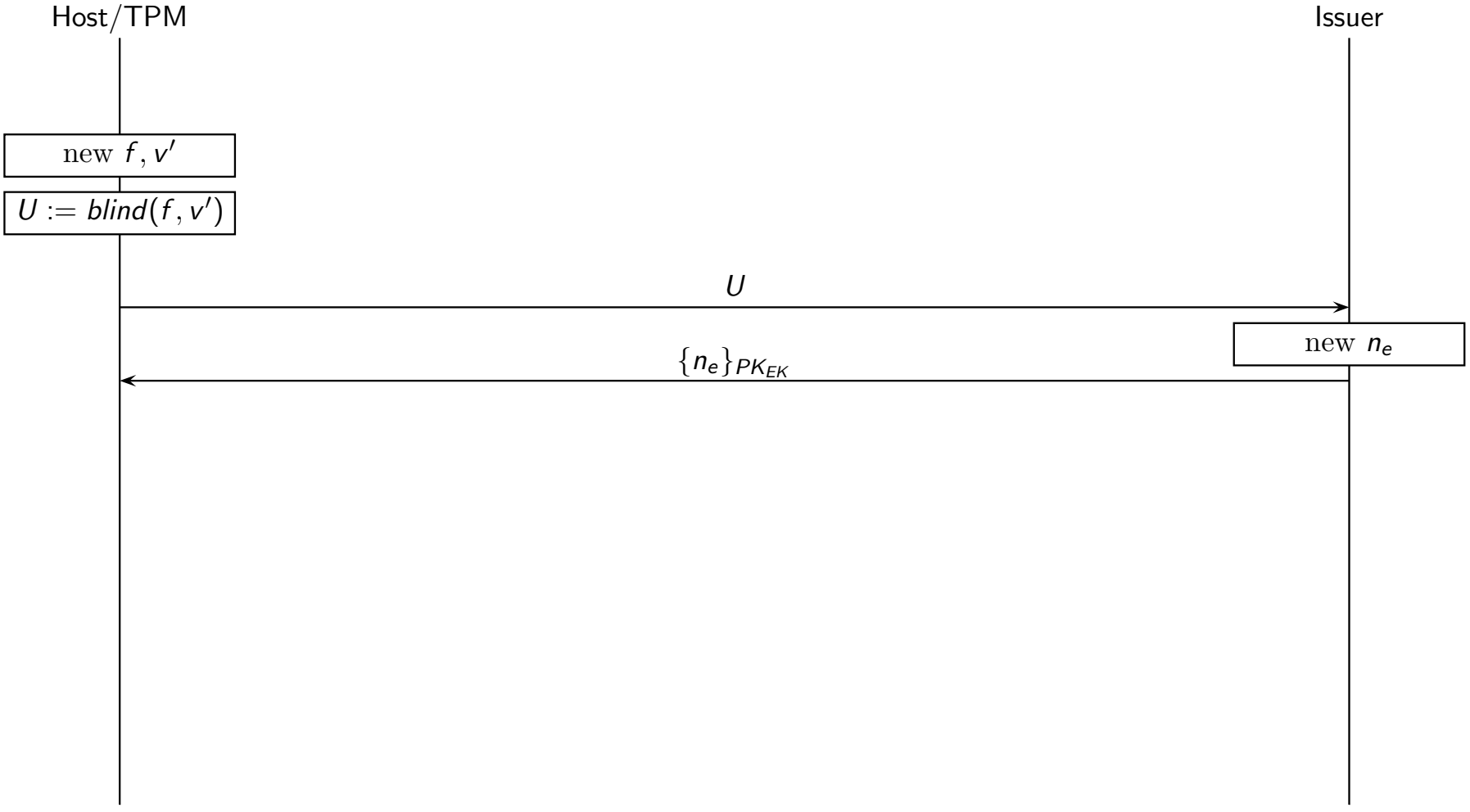
Join Protocol



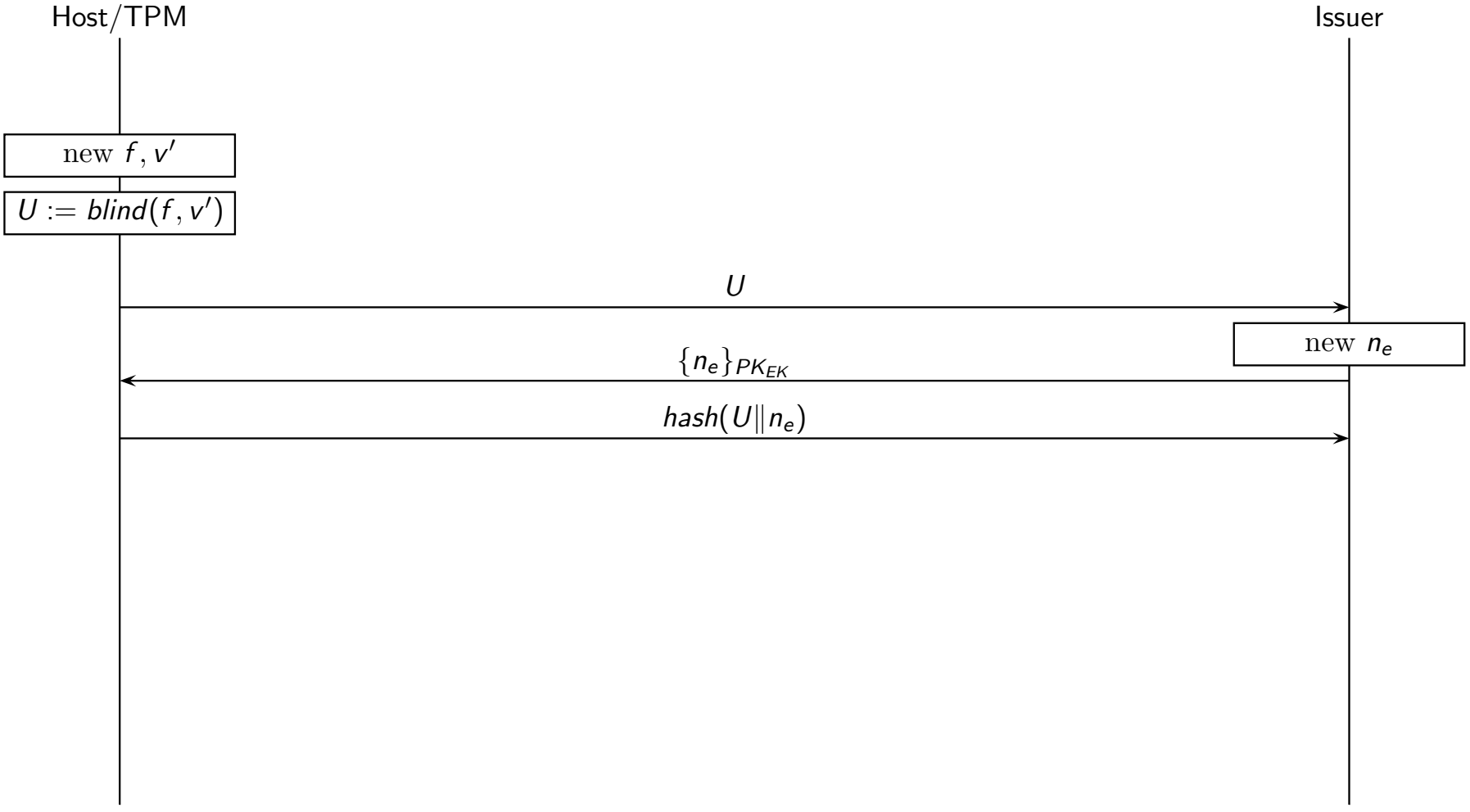
Join Protocol



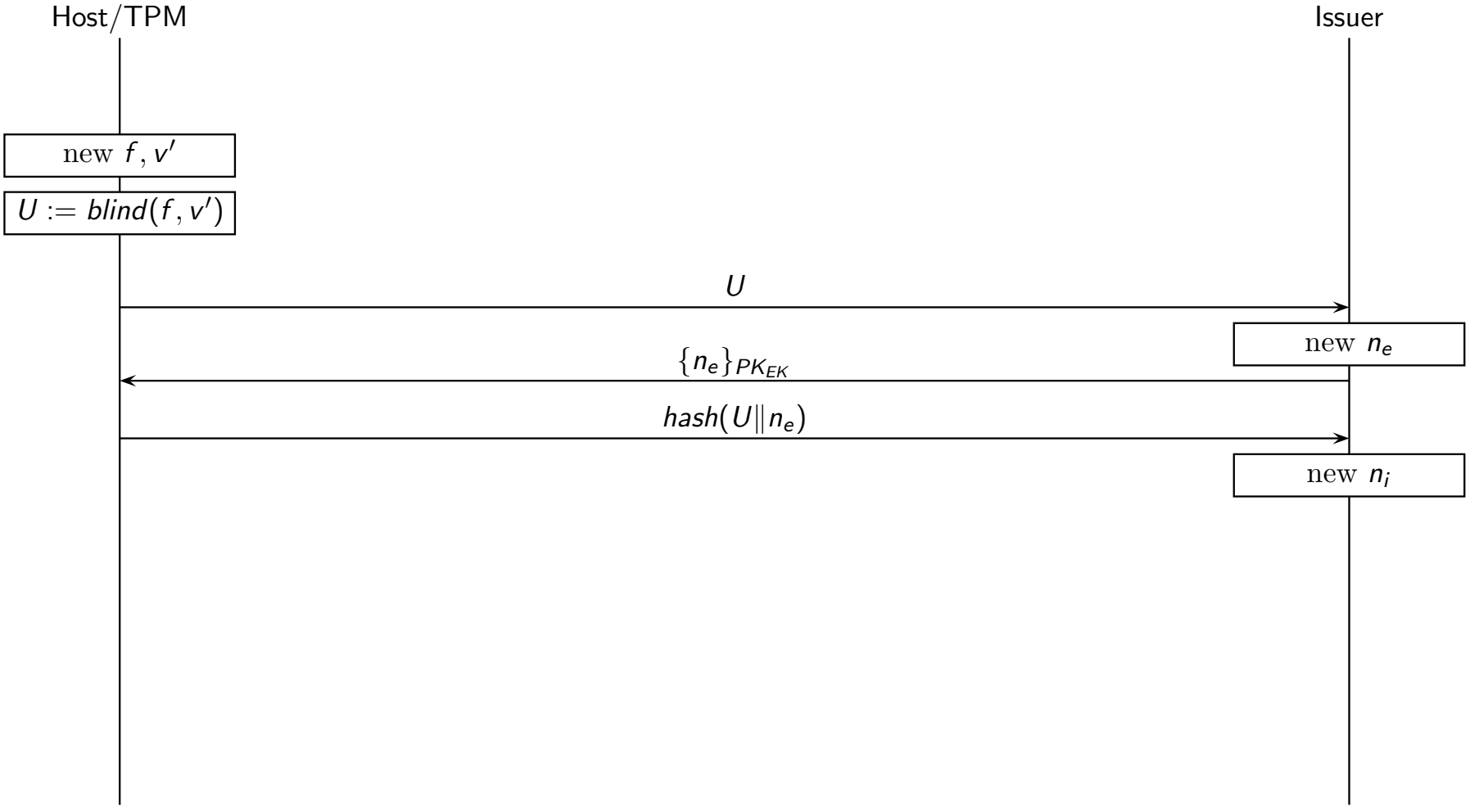
Join Protocol



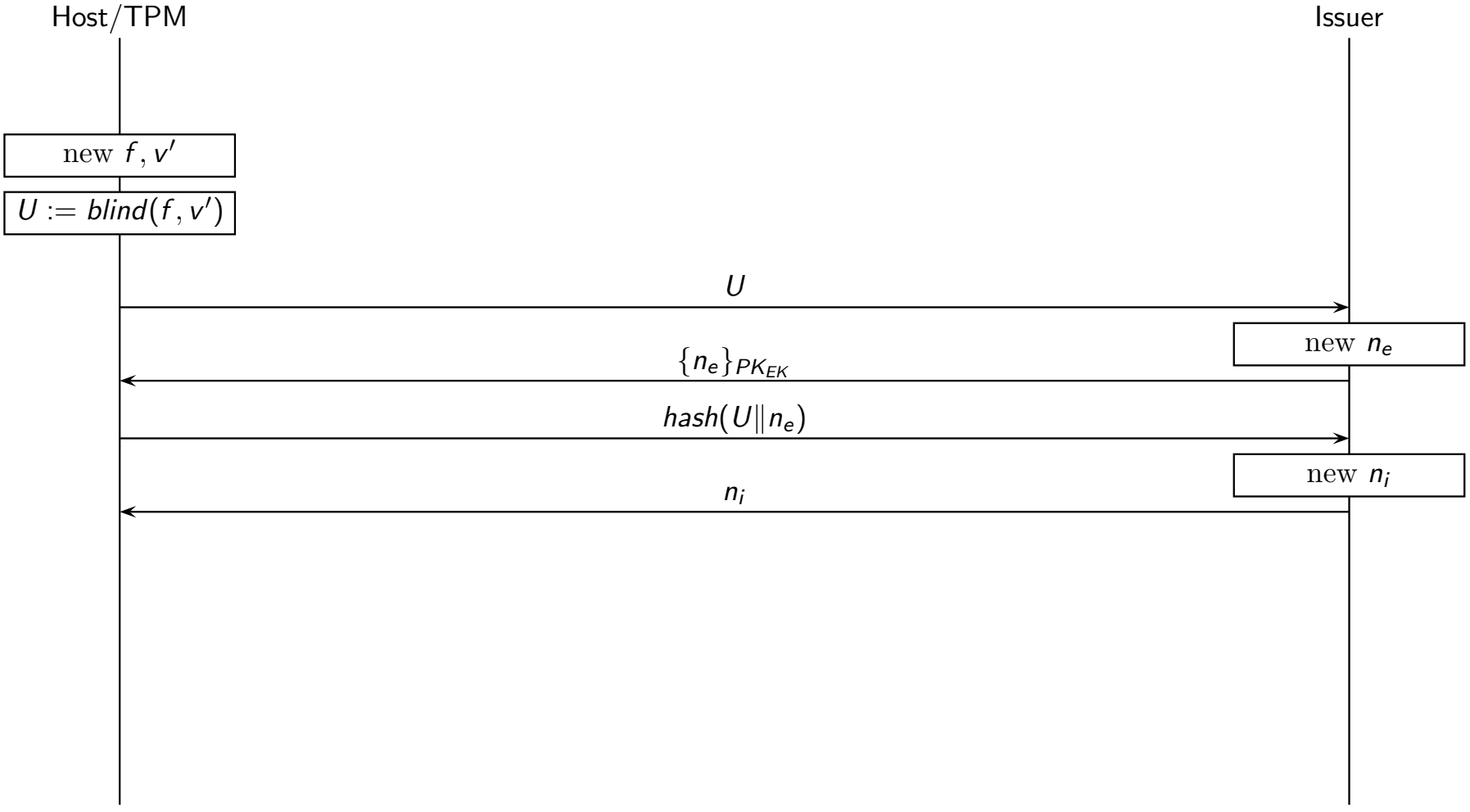
Join Protocol



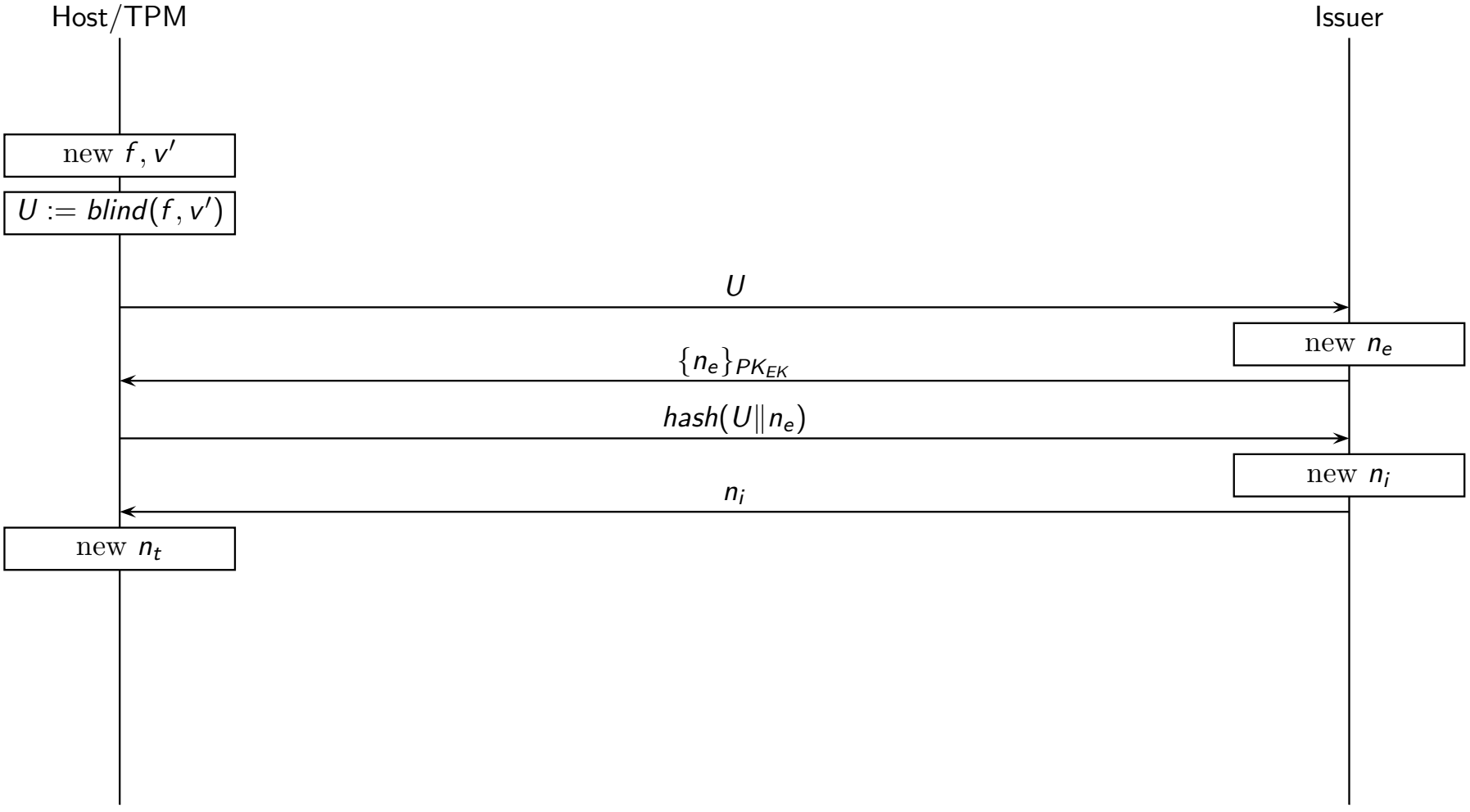
Join Protocol



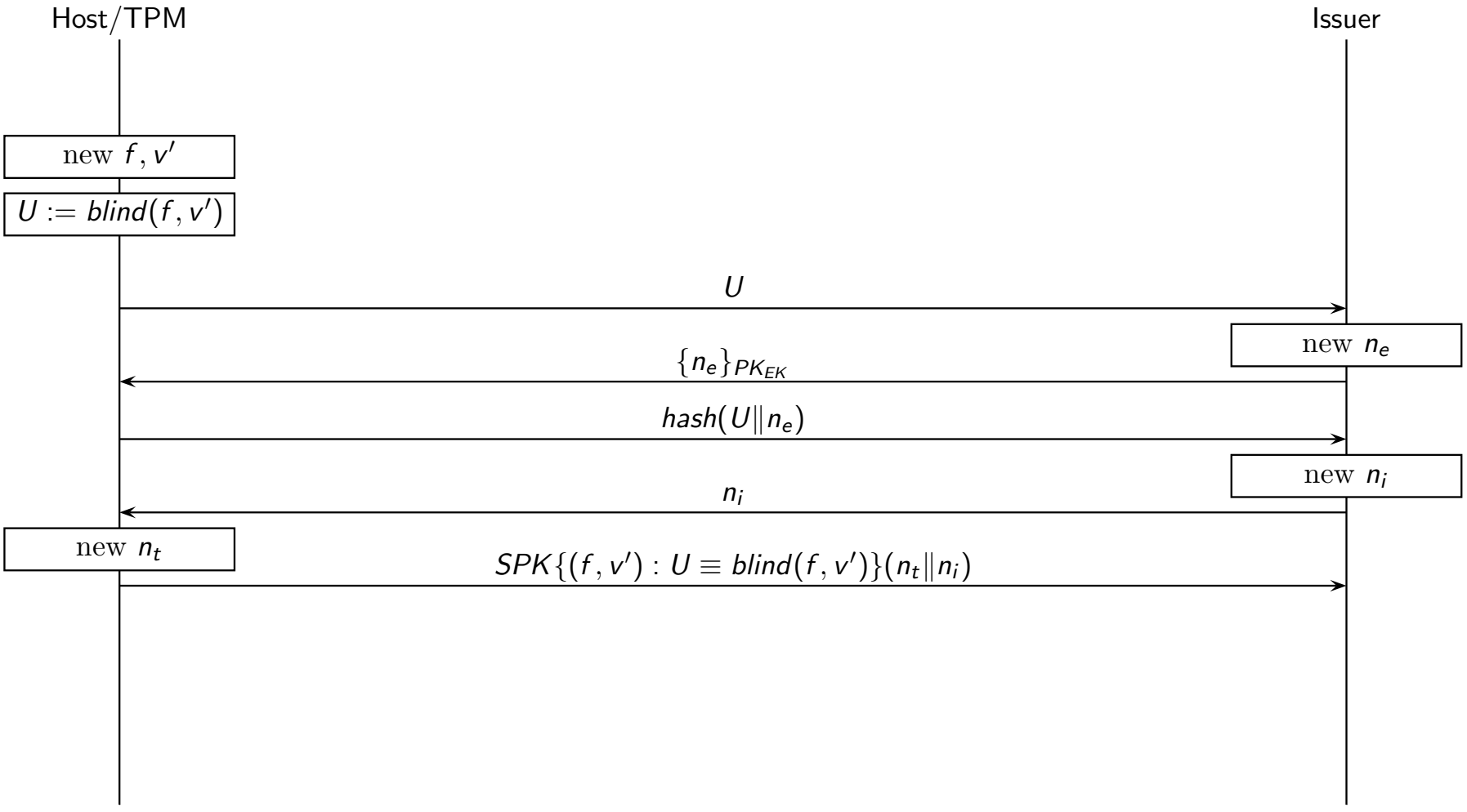
Join Protocol



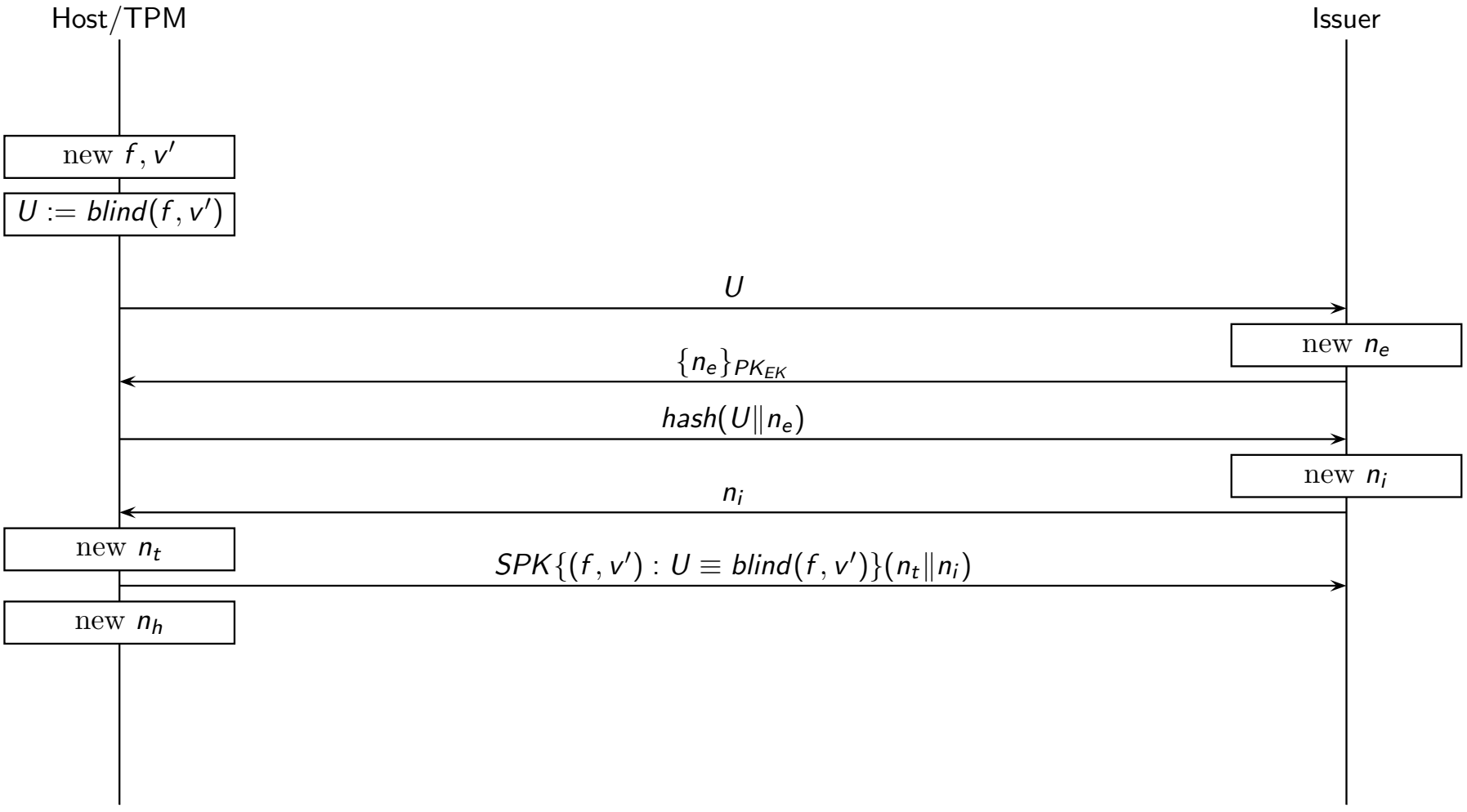
Join Protocol



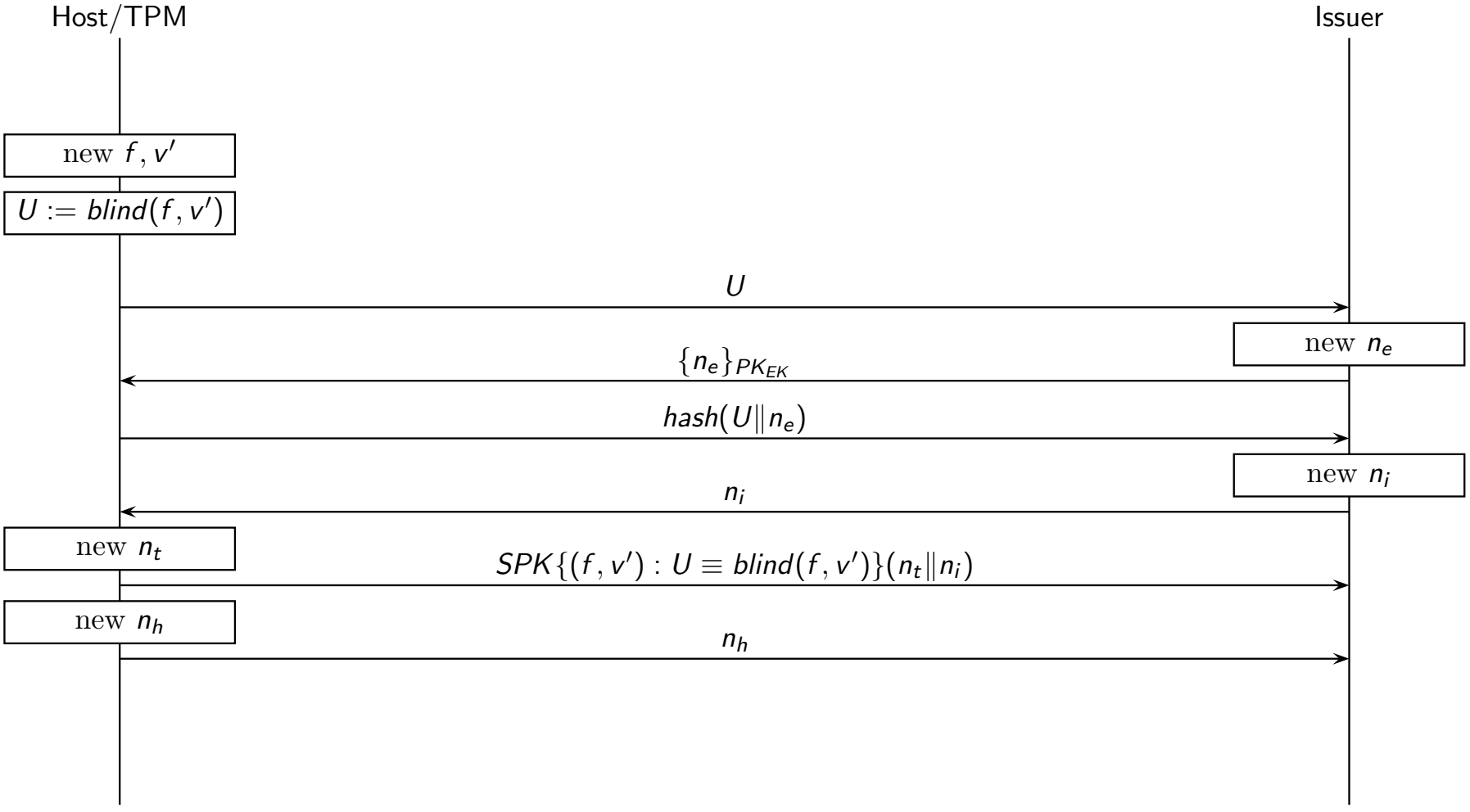
Join Protocol



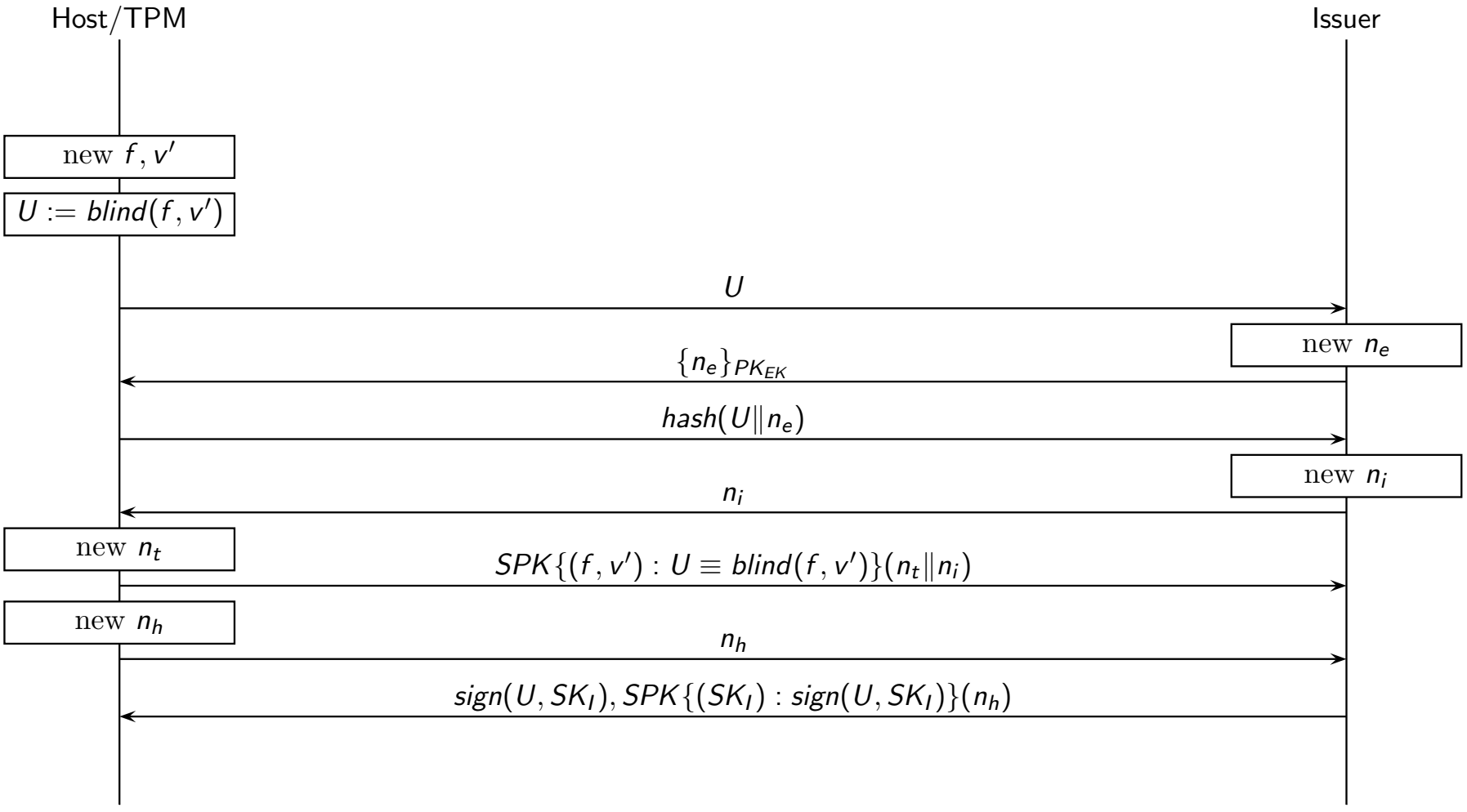
Join Protocol



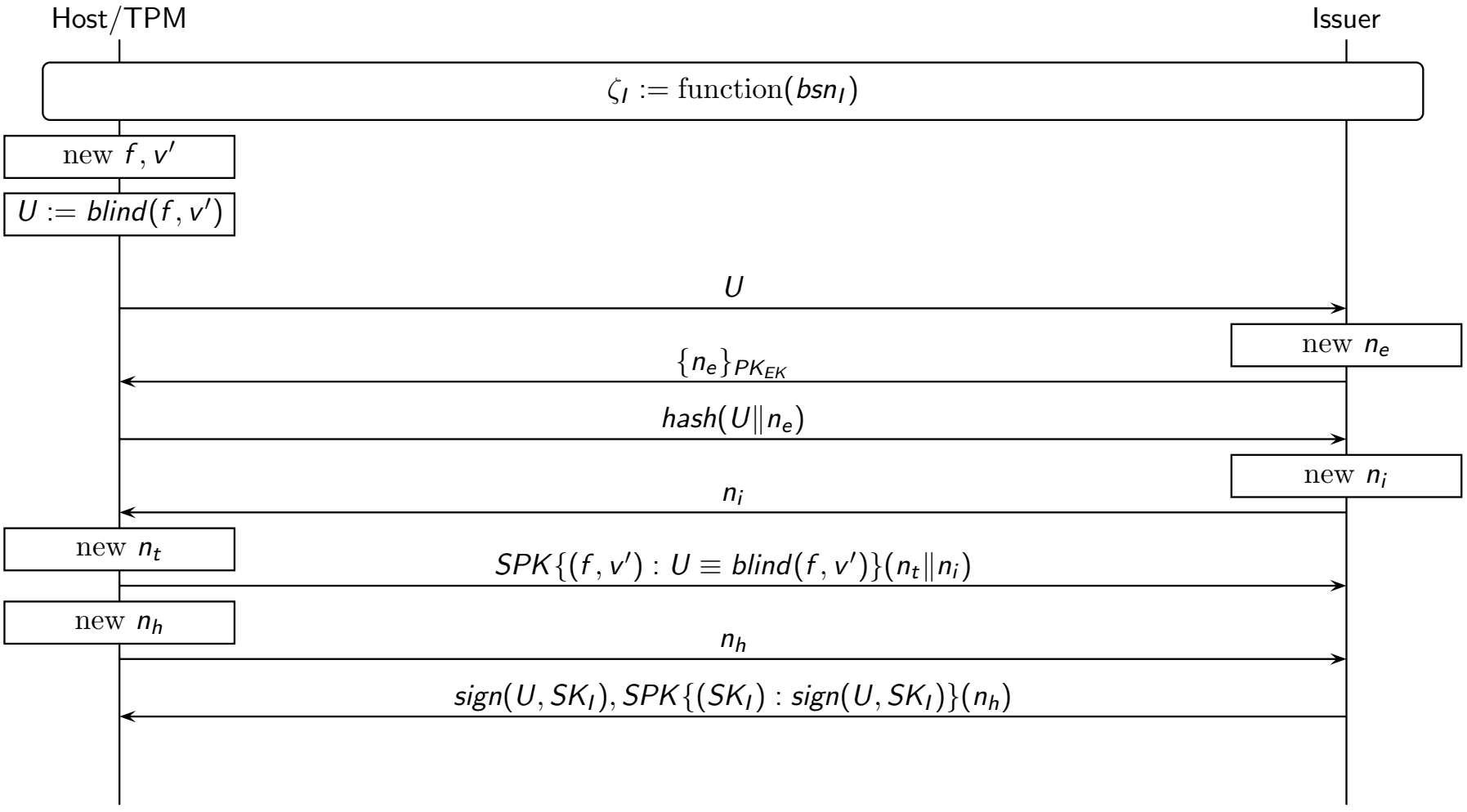
Join Protocol



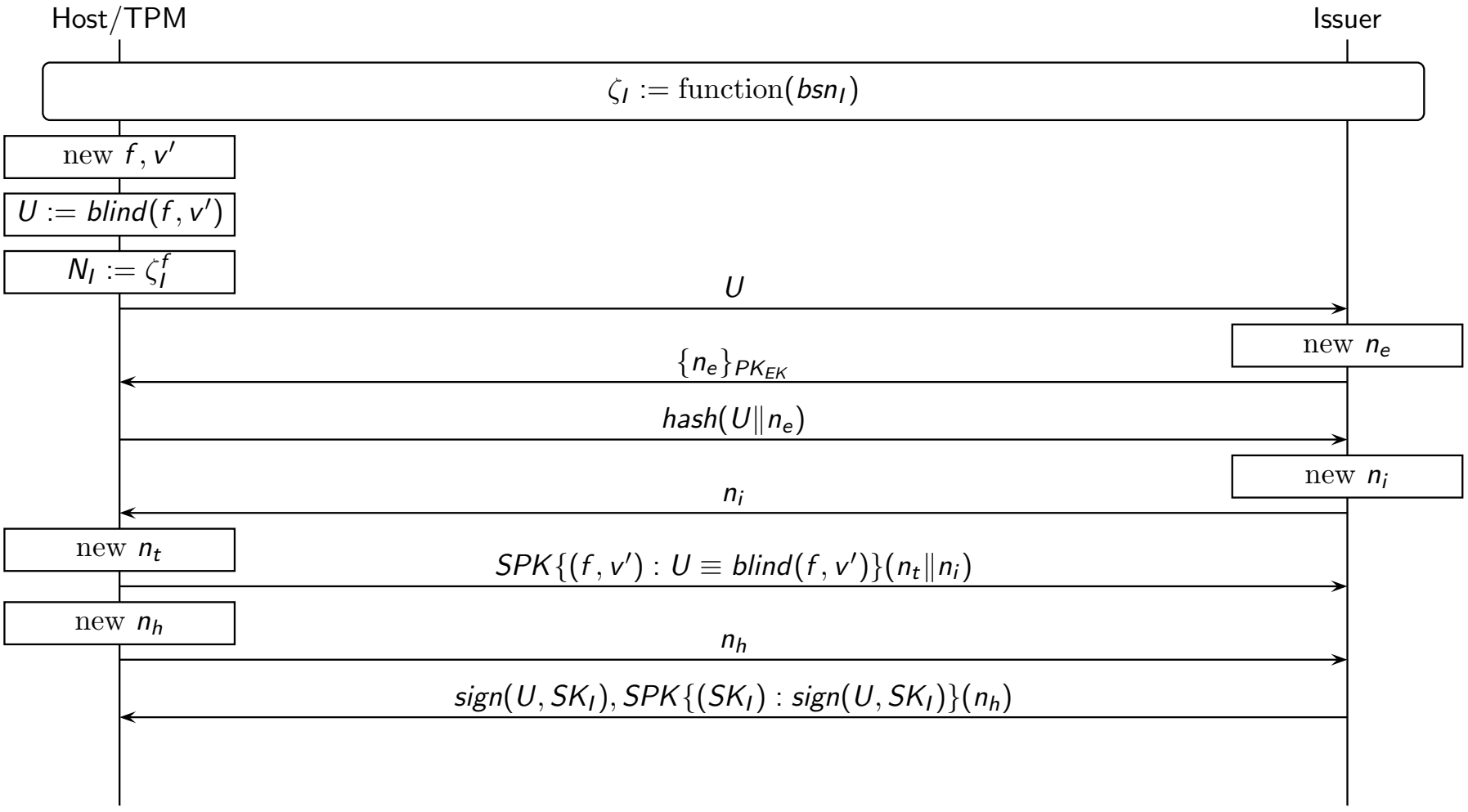
Join Protocol



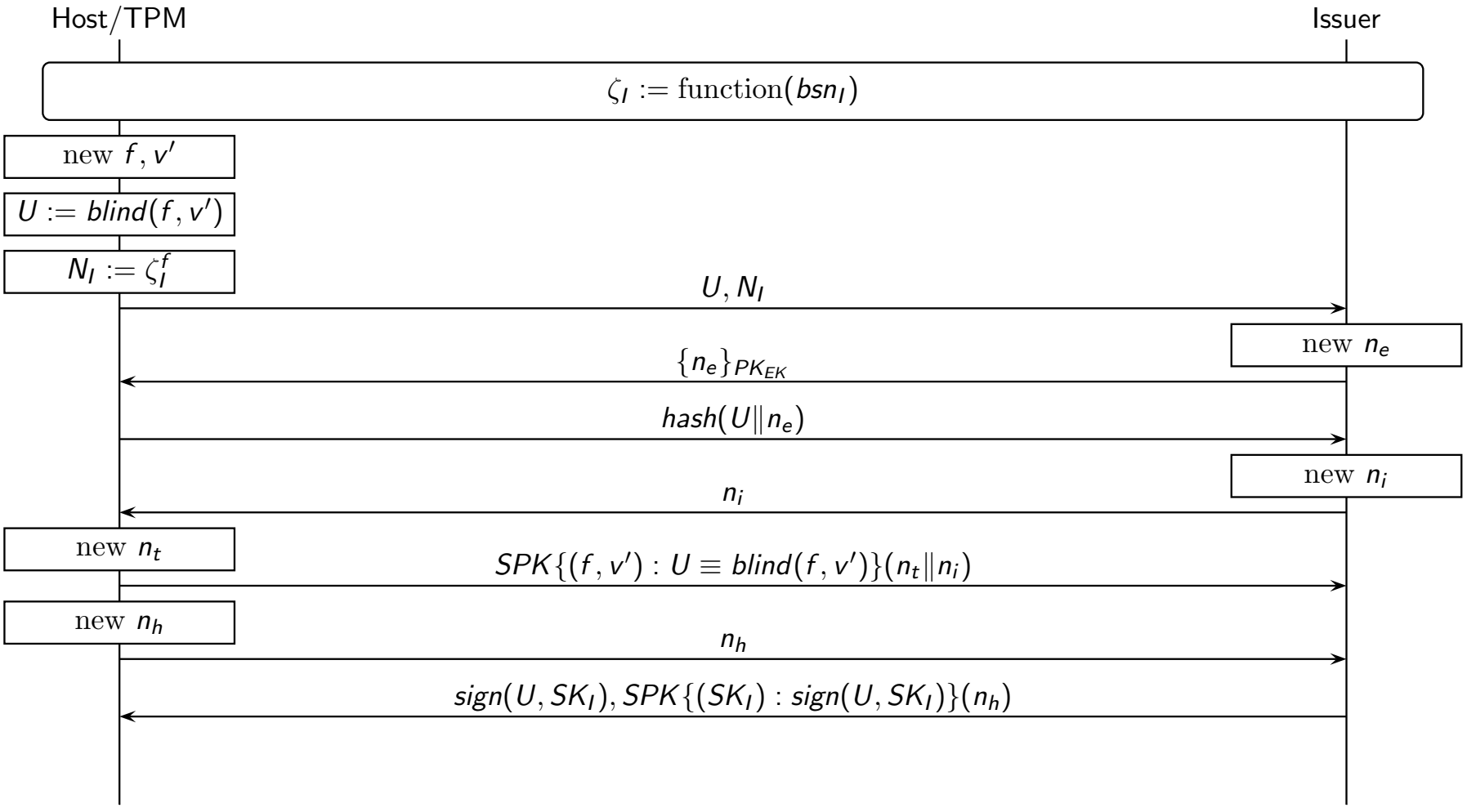
Join Protocol



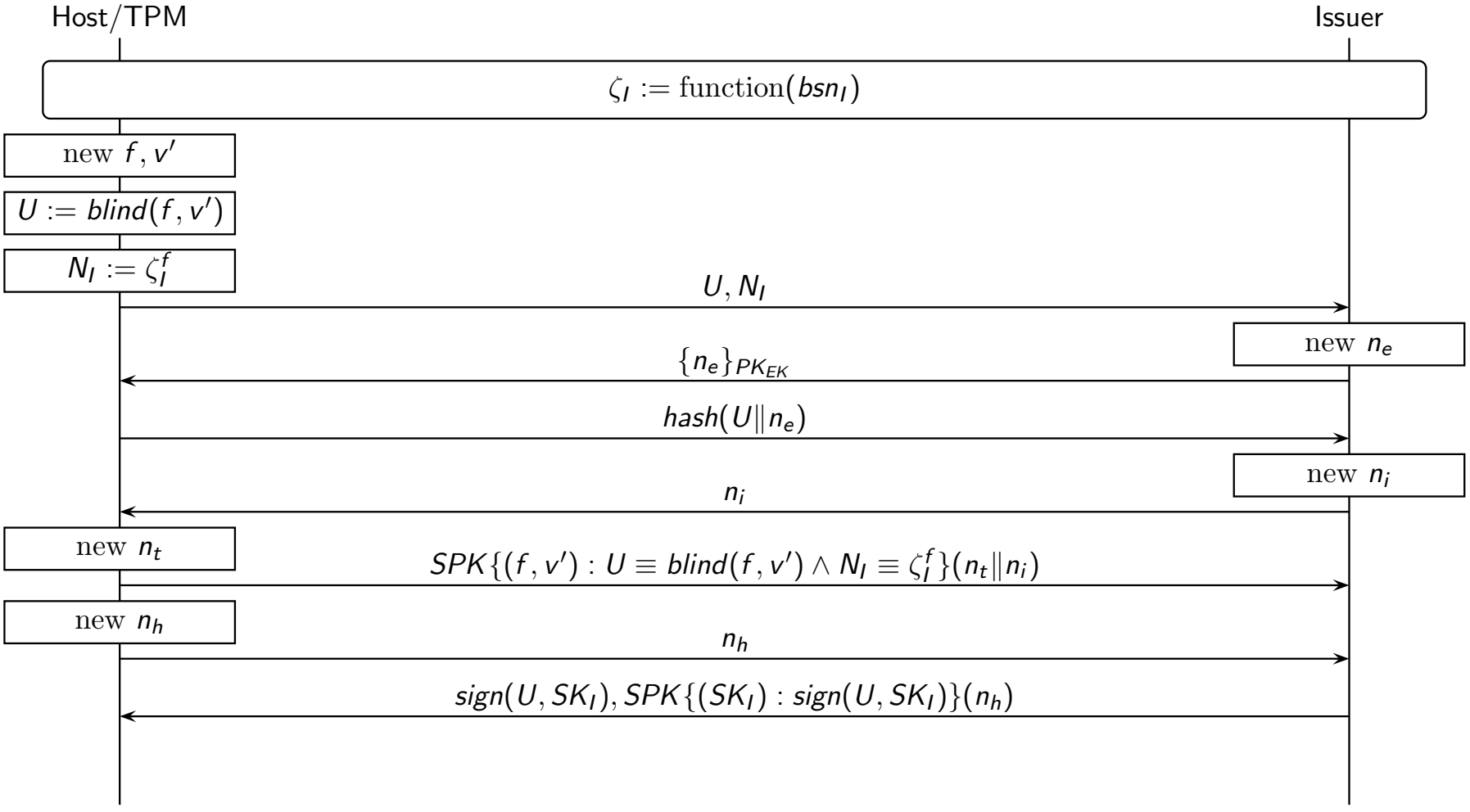
Join Protocol



Join Protocol

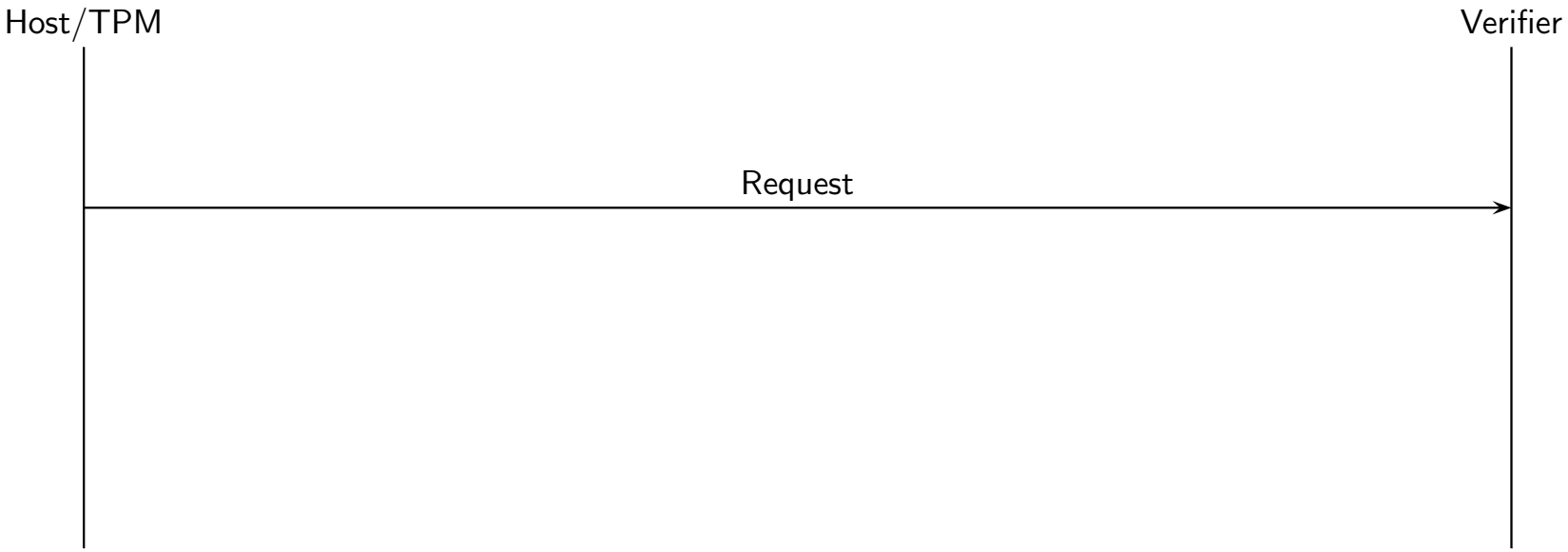


Join Protocol

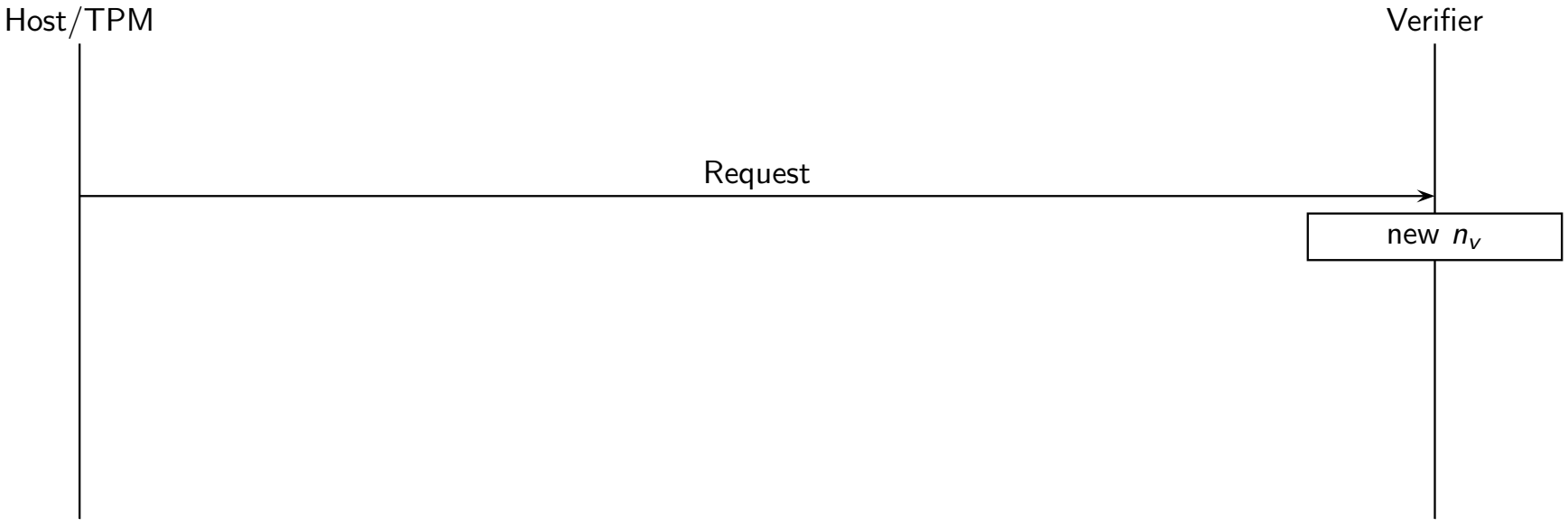


Sign/Verify Protocol

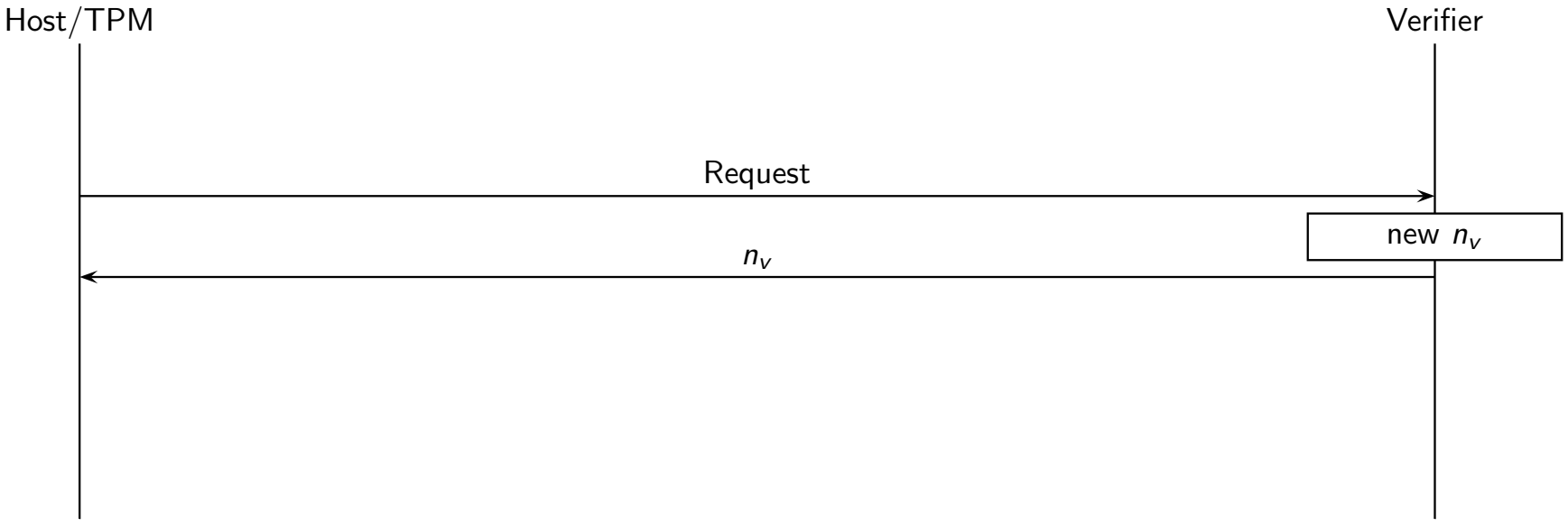
Sign/Verify Protocol



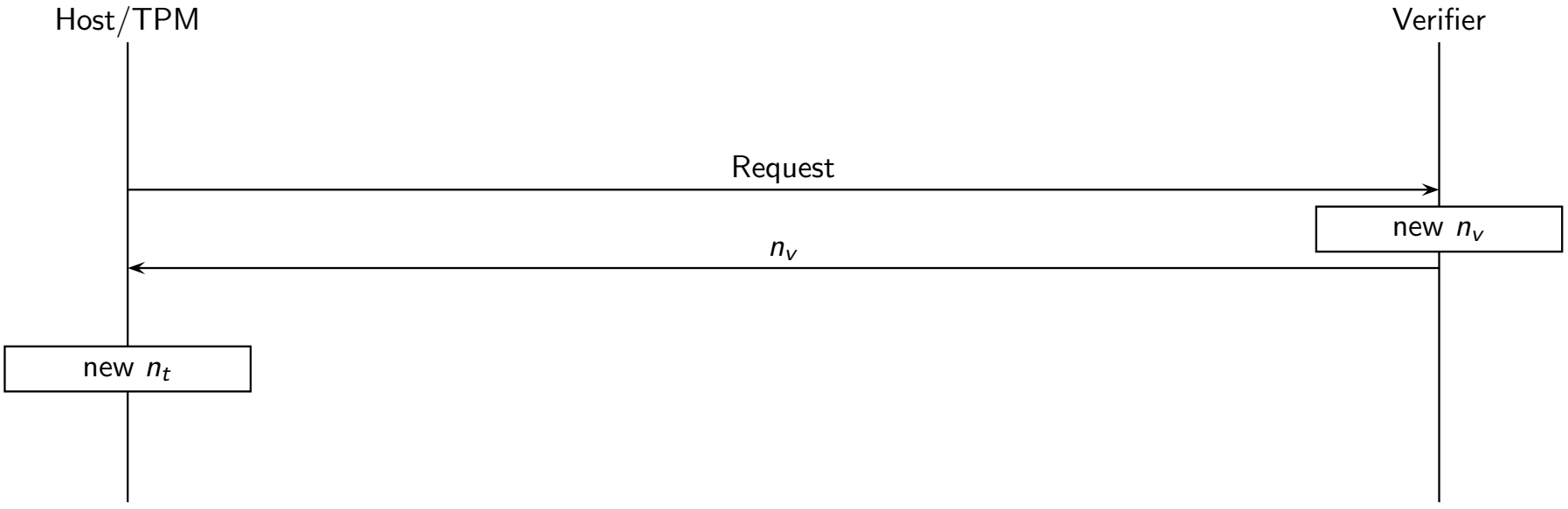
Sign/Verify Protocol



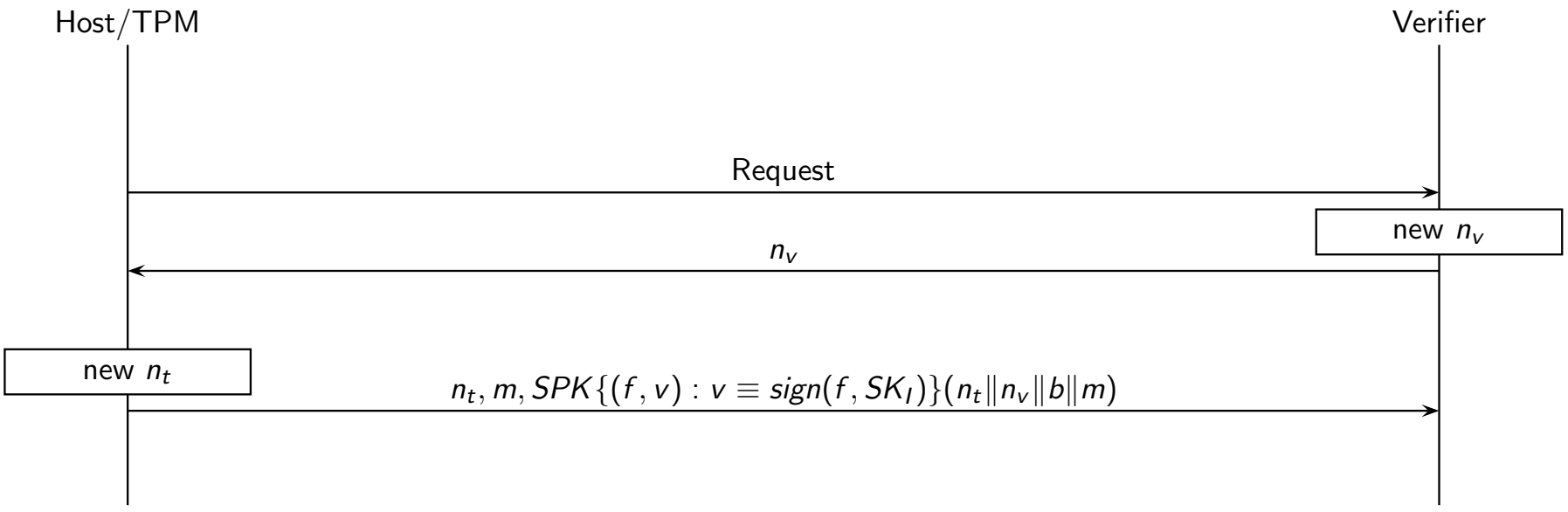
Sign/Verify Protocol



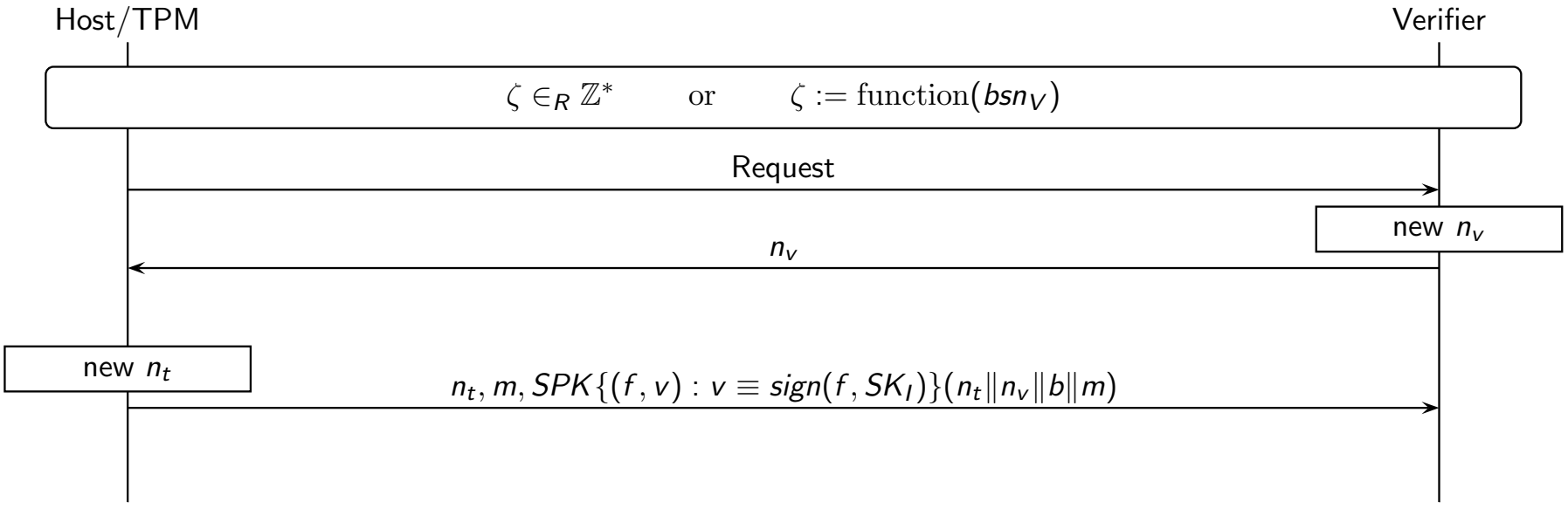
Sign/Verify Protocol



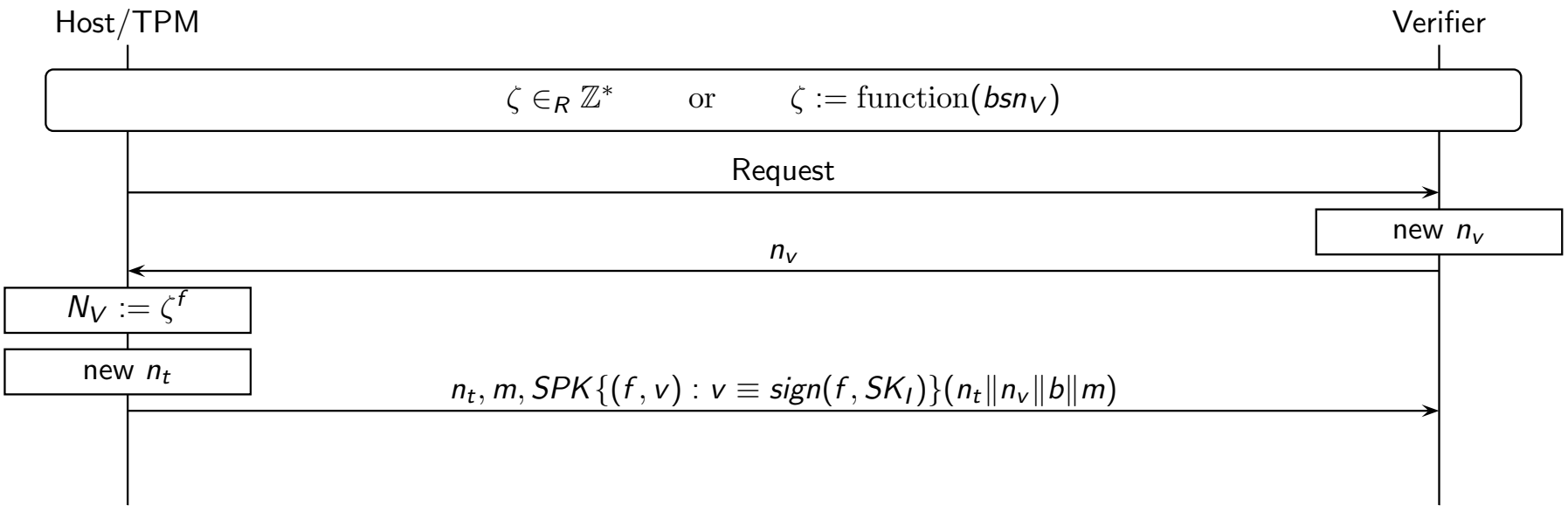
Sign/Verify Protocol



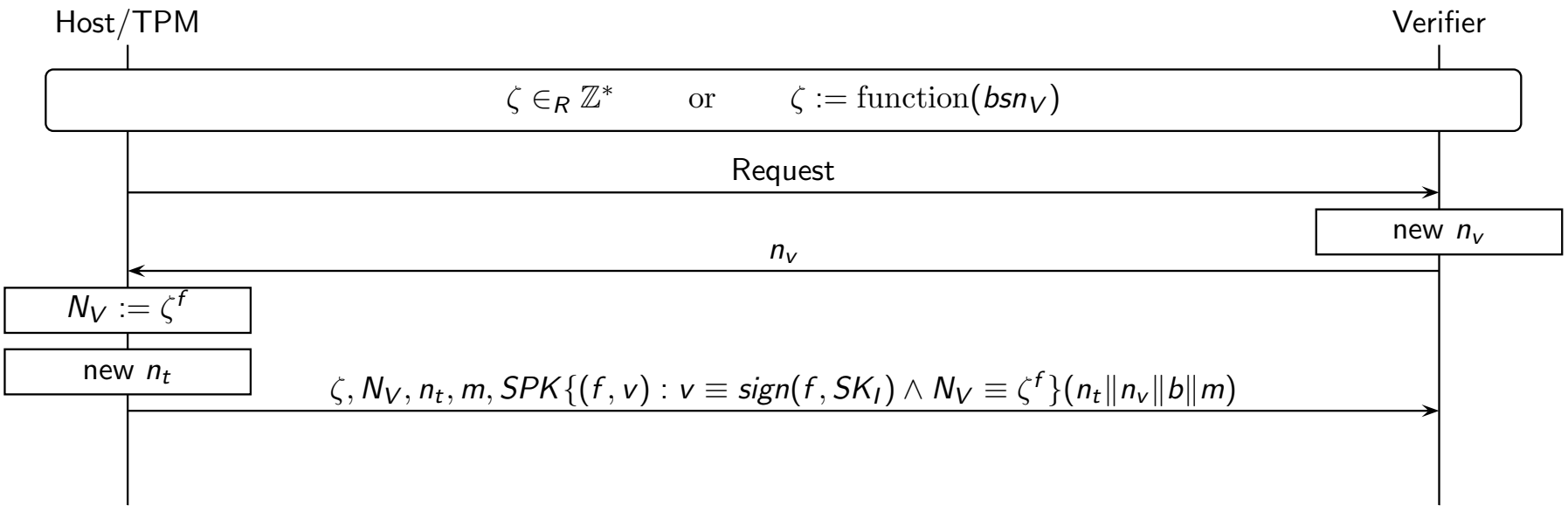
Sign/Verify Protocol



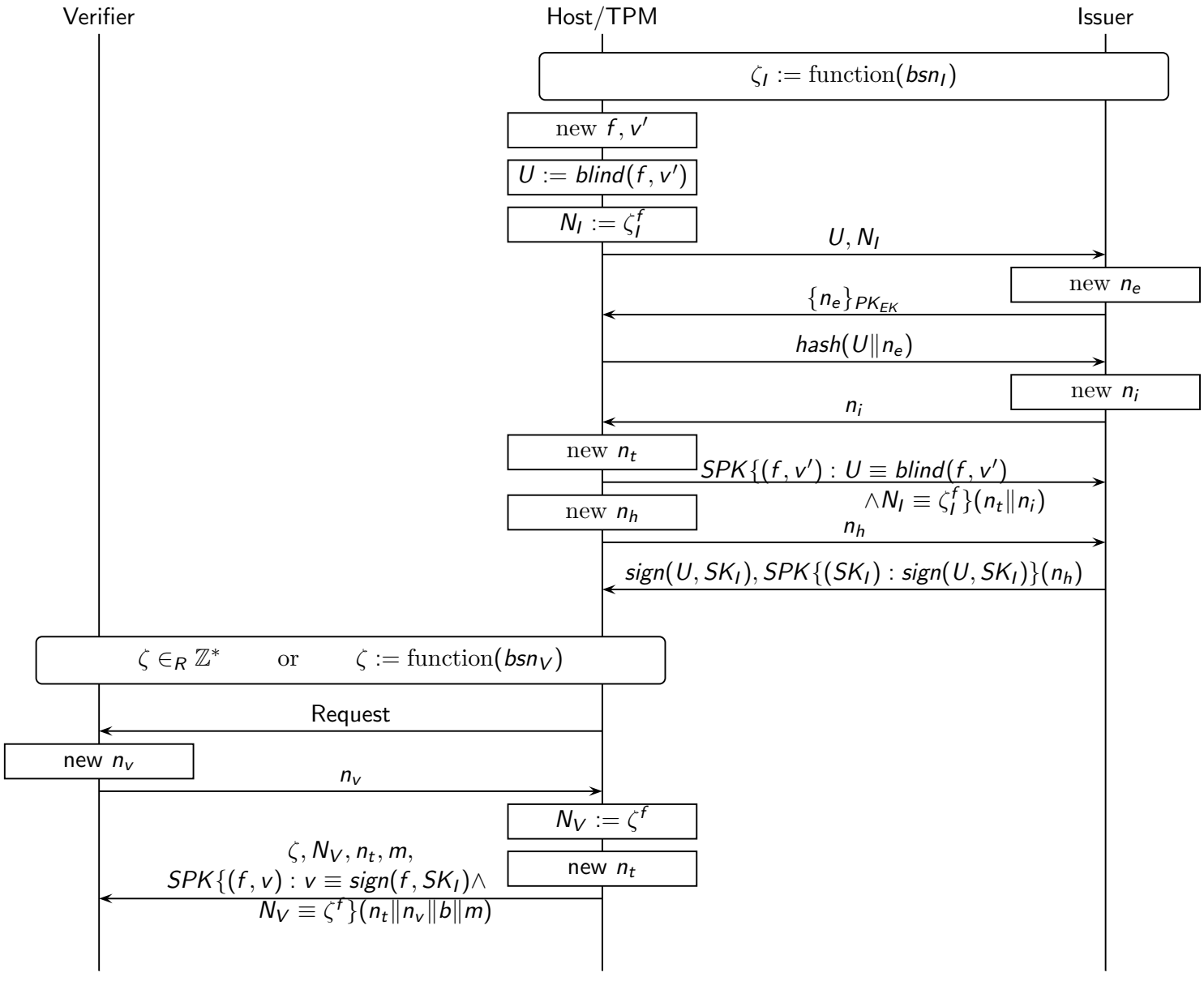
Sign/Verify Protocol



Sign/Verify Protocol



DAA Protocol



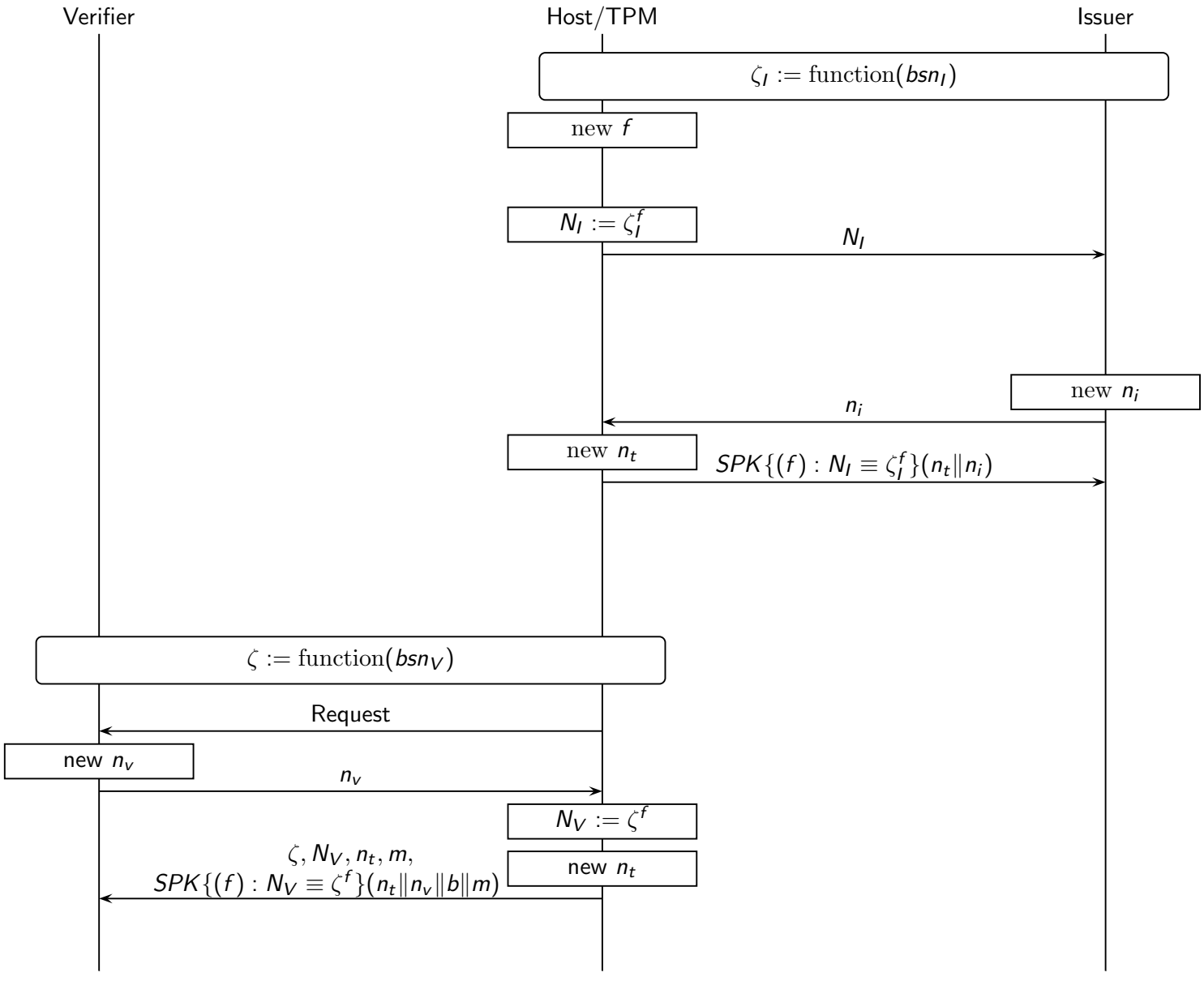
Violating privacy

Recall security properties

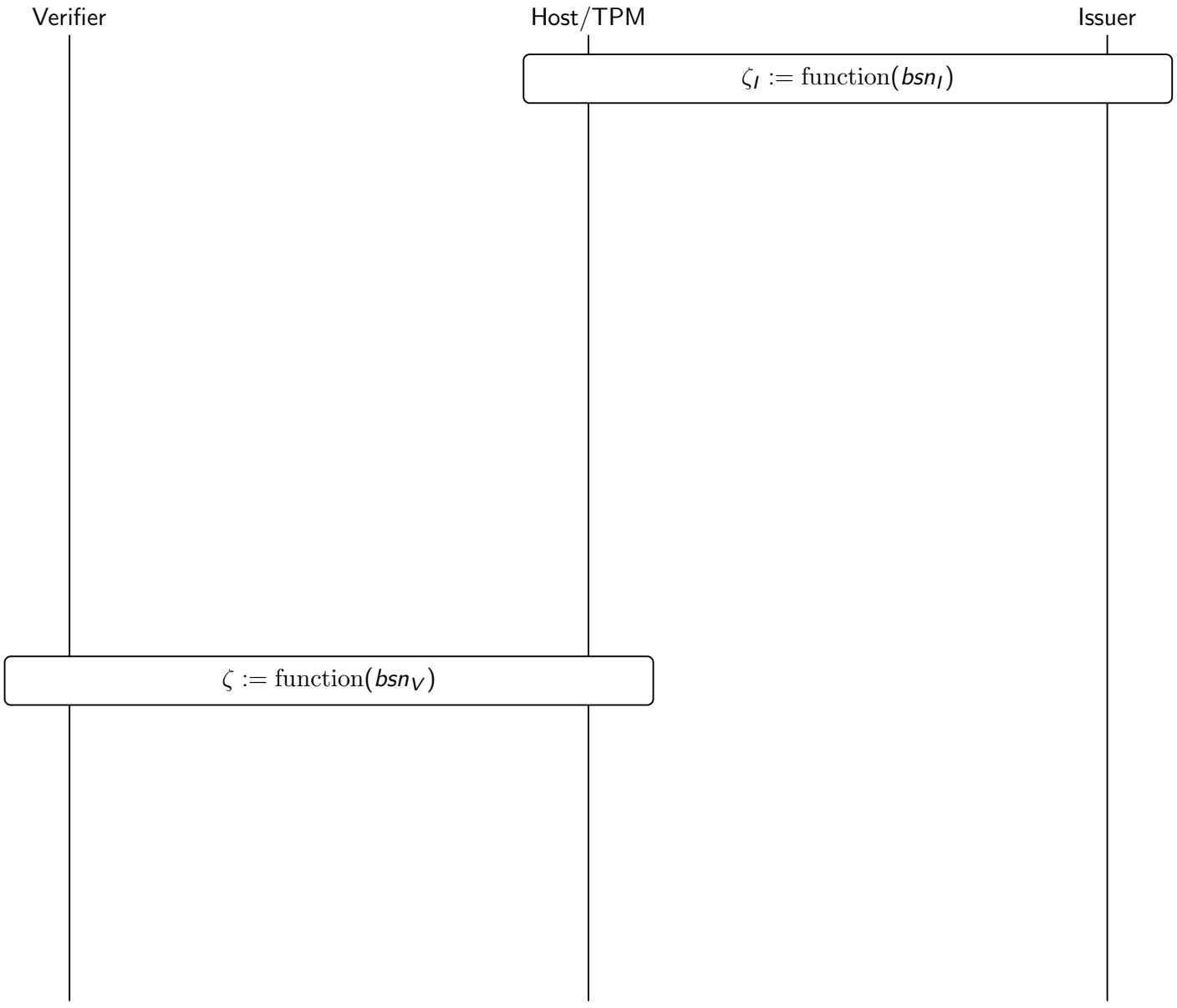
1. Only a trusted platform is able to authenticate
2. Privacy of a non-corrupt host is guaranteed by the sign/verify protocol:
   - 2.1 Interactions are anonymous
   - 2.2 Linkability (of transactions) is controlled by the user
3. Privacy is restored to a corrupted host if malicious software is removed

A colluding issuer and verifier can collude to violate security properties 2.1 & 2.2

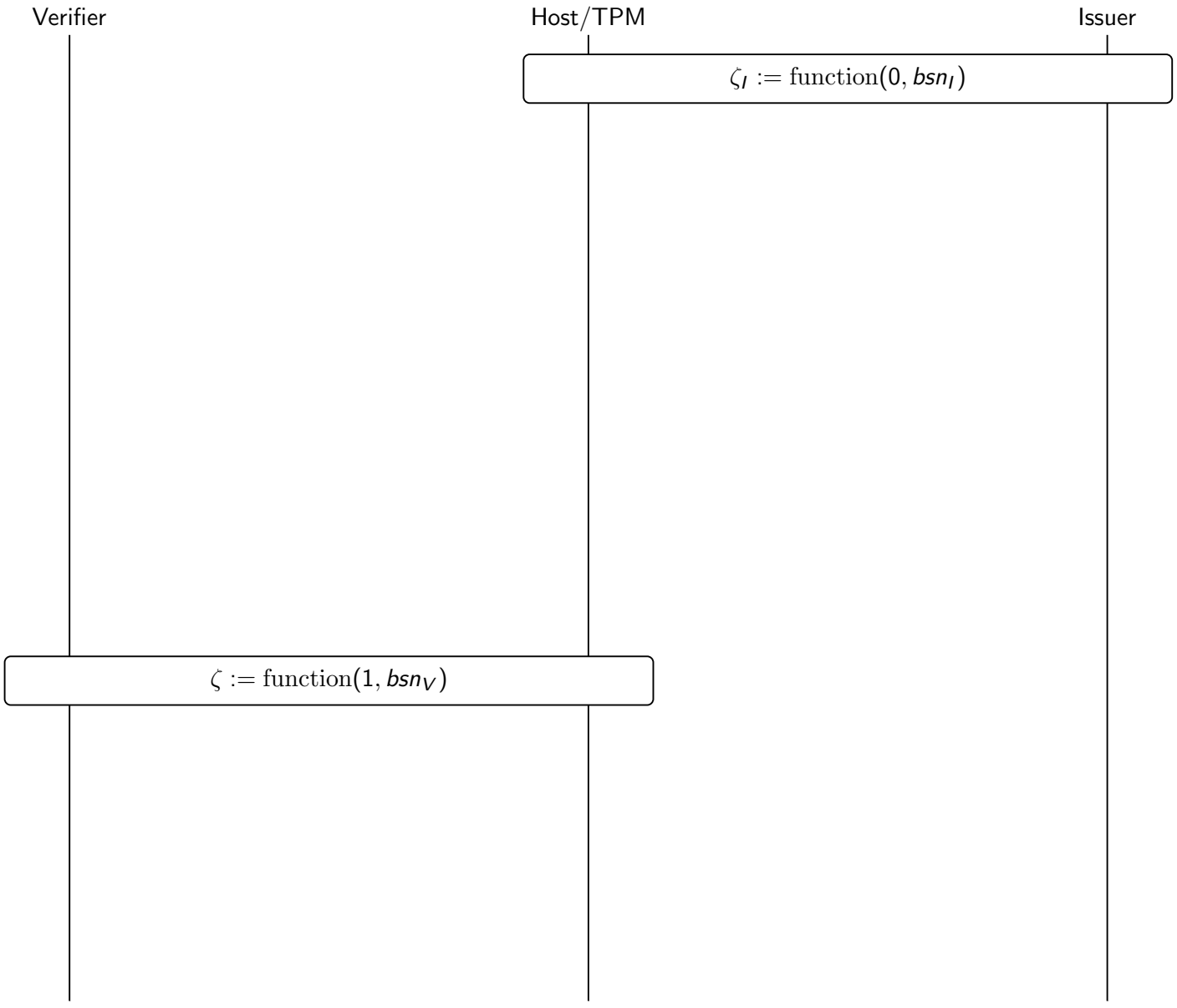
Violating privacy



Fix


Verifier
Host/TPM
Issuer
$\zeta_I := \text{function}(bsn_I)$
$\zeta := \text{function}(bsn_V)$

Fix


Verifier
Host/TPM
Issuer
$\zeta_I := \text{function}(0, bsn_I)$
$\zeta := \text{function}(1, bsn_V)$

Overcoming problems with DAA basenames

... Need to write these slides ....

... May not include them in presentation - depends on time ....

Conclusion/further work

- Identified a weakness in the DAA protocol
- Provided a fix which requires minimal alteration
- The fix does not require any changes to the hardware TPM

- Simpler solutions?
- Finer-grained approach to privacy
- Development of formal approaches for the verification of protocols such as DAA