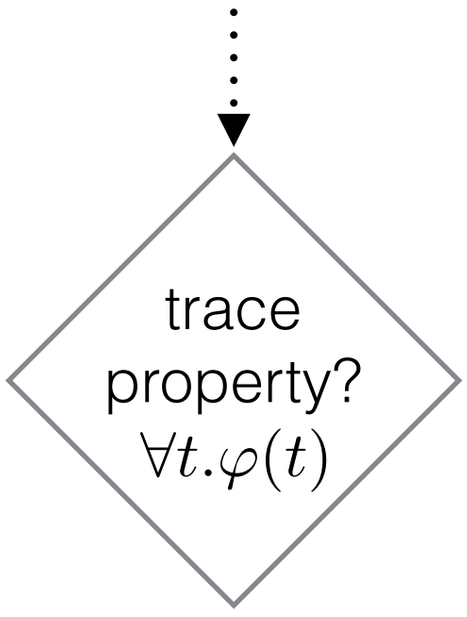
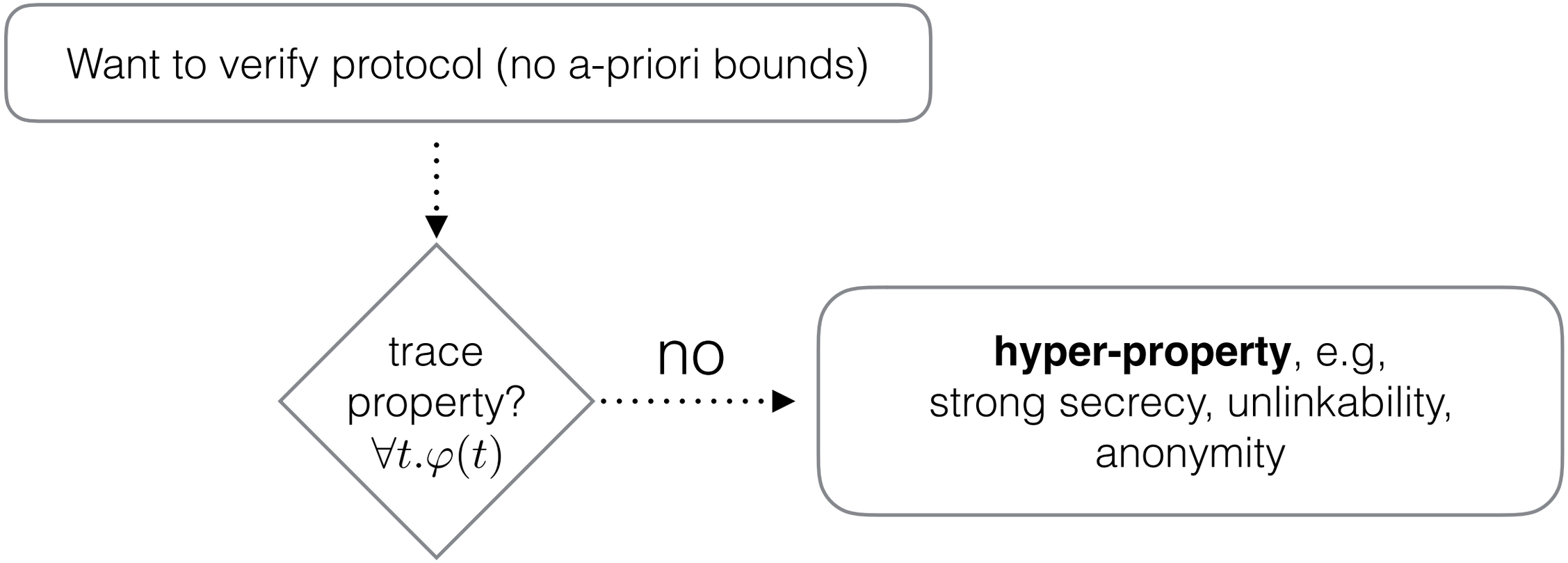
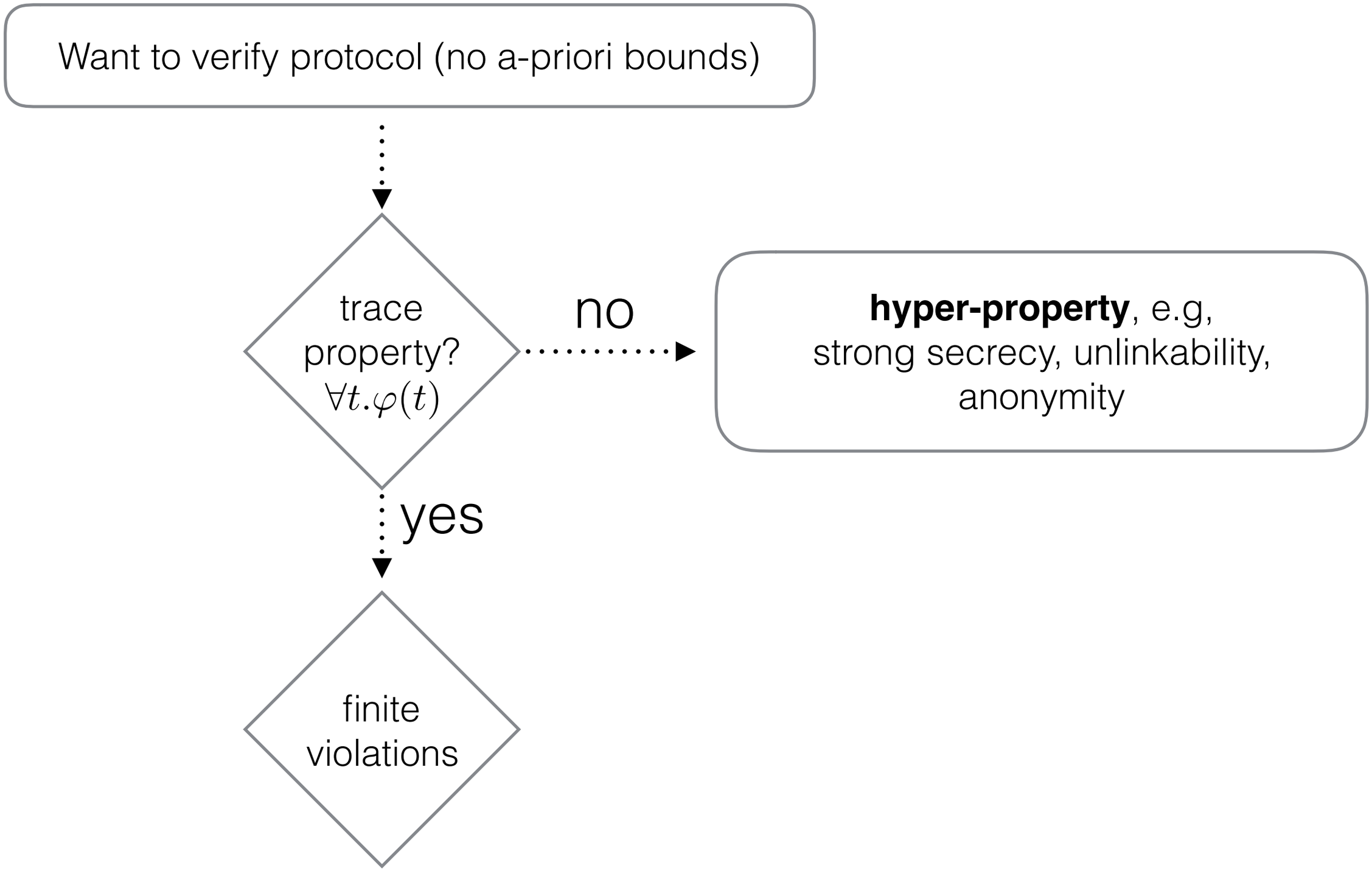
A Novel Approach for Reasoning about Liveness in Cryptographic Protocols and its Application to Fair Exchange

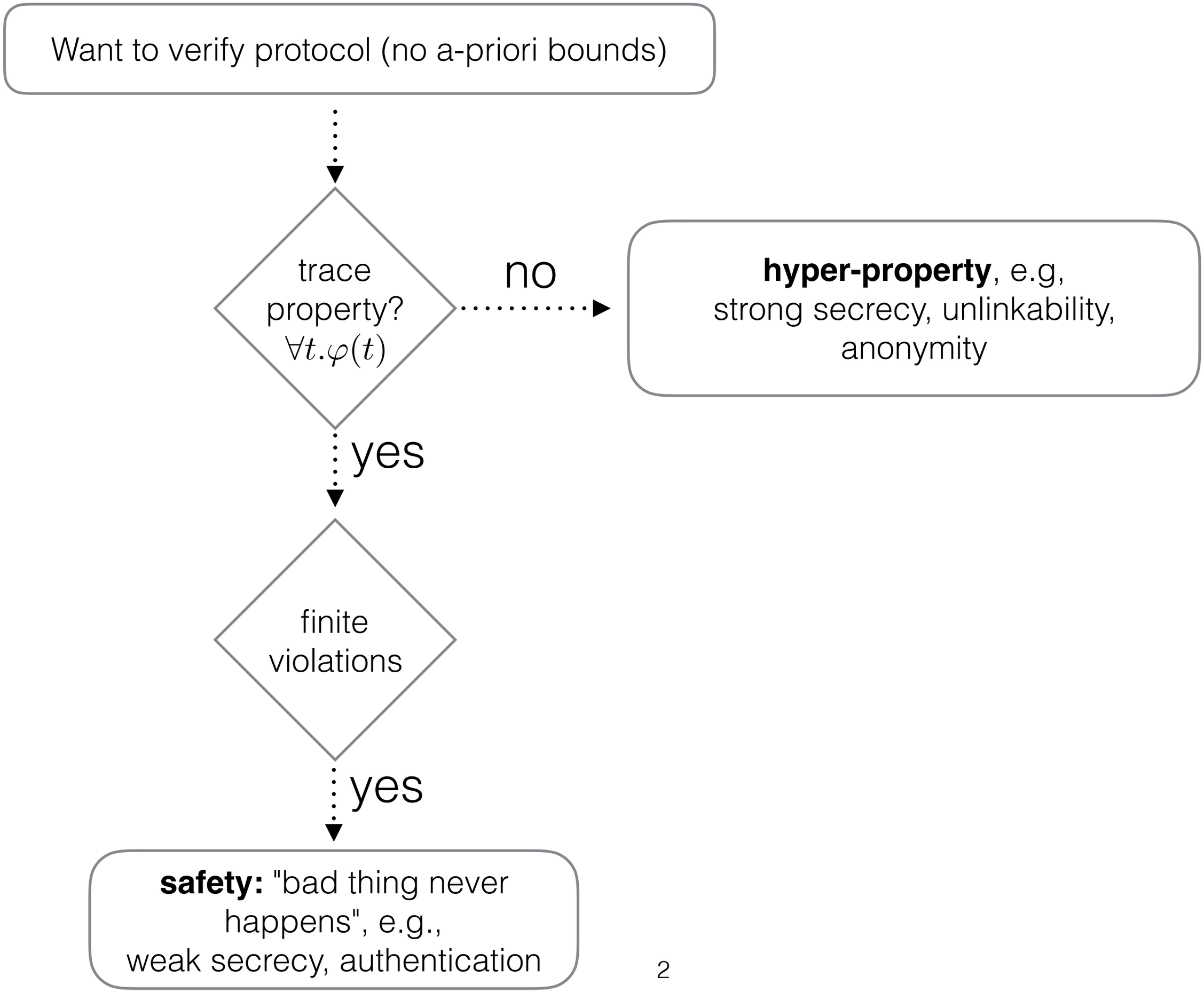
Michael Backes (CISPA, Saarland University, MPI-SWS), Jannik Dreier, Steve Kremer (LORIA, INRIA, CNRS, Université de Lorraine) and Robert Künnemann (CISPA, Saarland University)

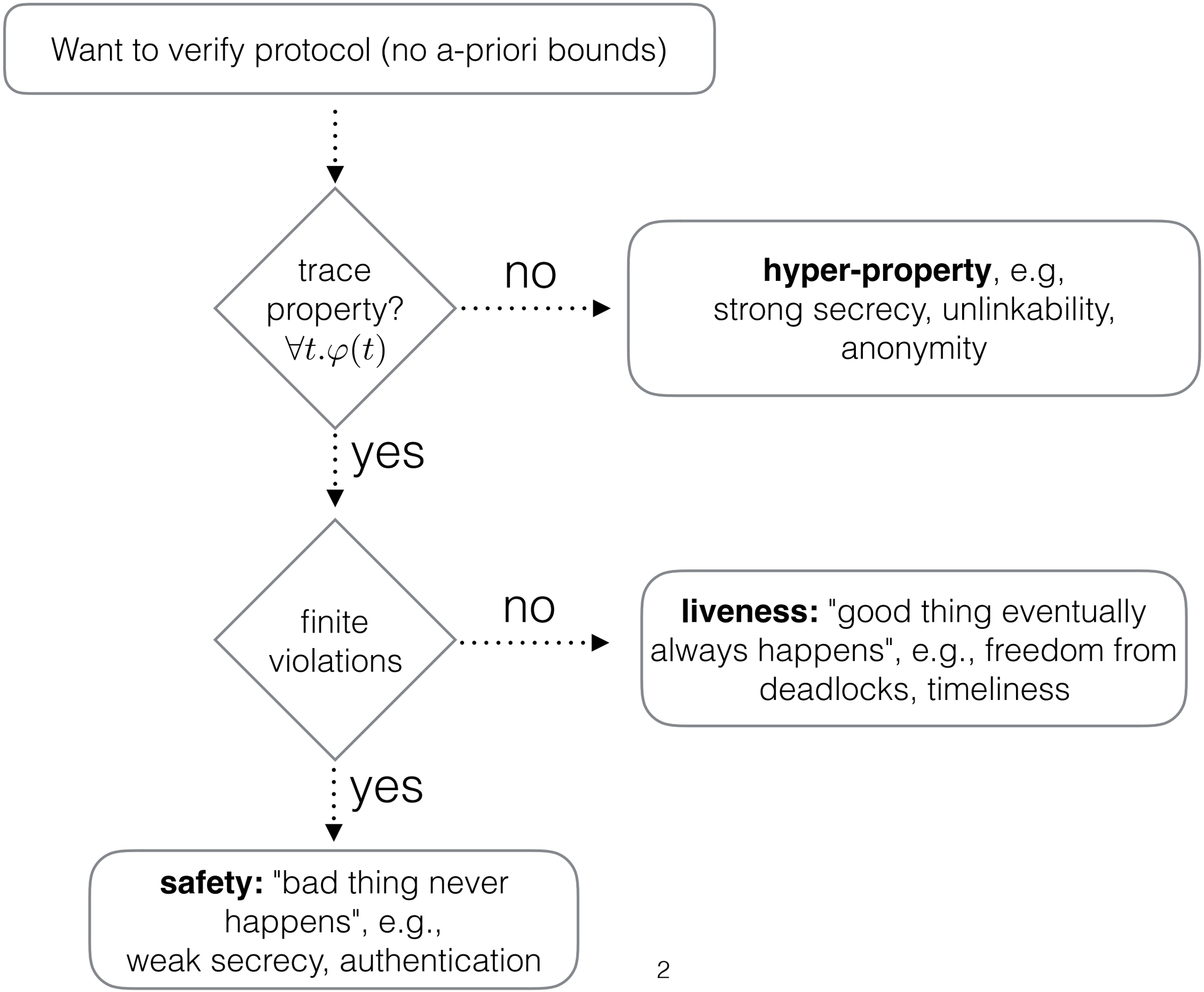
Want to verify protocol (no a-priori bounds)




Want to verify protocol (no a-priori bounds)
trace property? $\forall t.\varphi(t)$
no
hyper-property, e.g, strong secrecy, unlinkability, anonymity


Want to verify protocol (no a-priori bounds)
trace property? $\forall t.\varphi(t)$
no
hyper-property, e.g, strong secrecy, unlinkability, anonymity
yes
finite violations


Want to verify protocol (no a-priori bounds)
trace property? $\forall t.\varphi(t)$
no
hyper-property, e.g, strong secrecy, unlinkability, anonymity
yes
finite violations
yes
safety: "bad thing never happens", e.g., weak secrecy, authentication


Want to verify protocol (no a-priori bounds)
trace property? $\forall t.\varphi(t)$
no
hyper-property, e.g, strong secrecy, unlinkability, anonymity
yes
finite violations
no
liveness: "good thing eventually always happens", e.g., freedom from deadlocks, timeliness
yes
safety: "bad thing never happens", e.g., weak secrecy, authentication

When do you need liveness?

- when "no response" is dangerous, e.g., industrial control systems

- when recovery is a goal, e.g., self-healing crypto implementations

- when revocation or fairness are (sub)goals, e.g., accountability mechanisms

(c) EDF

When do you need liveness?

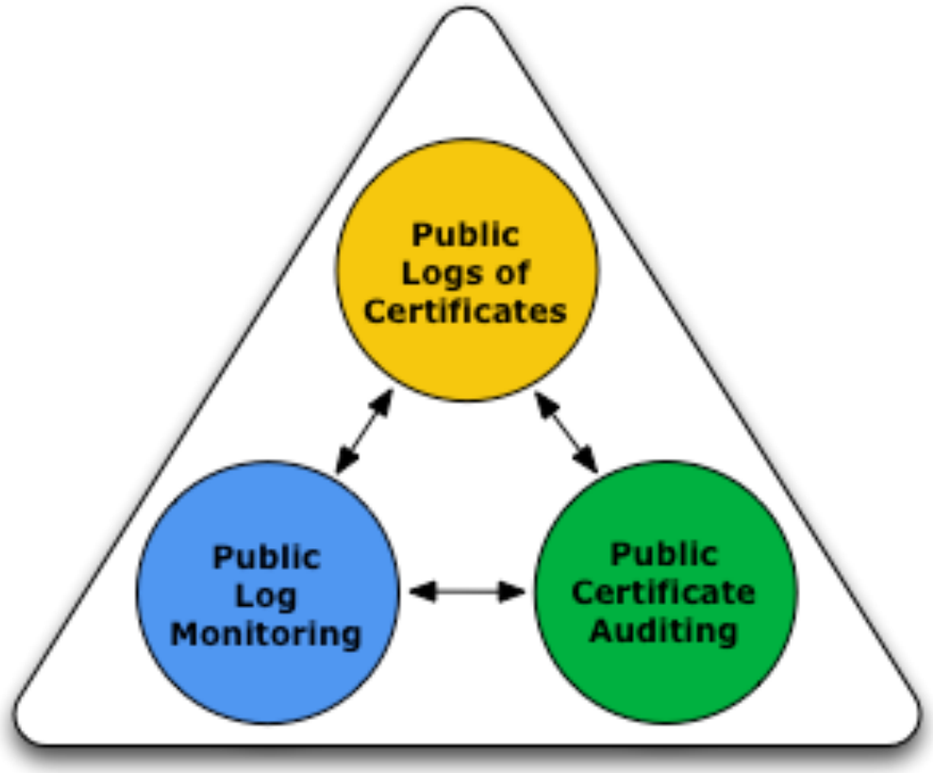
- when "no response" is dangerous, e.g., industrial control systems
- when recovery is a goal, e.g., self-healing crypto implementations
- when revocation or fairness are (sub)goals, e.g., accountability mechanisms

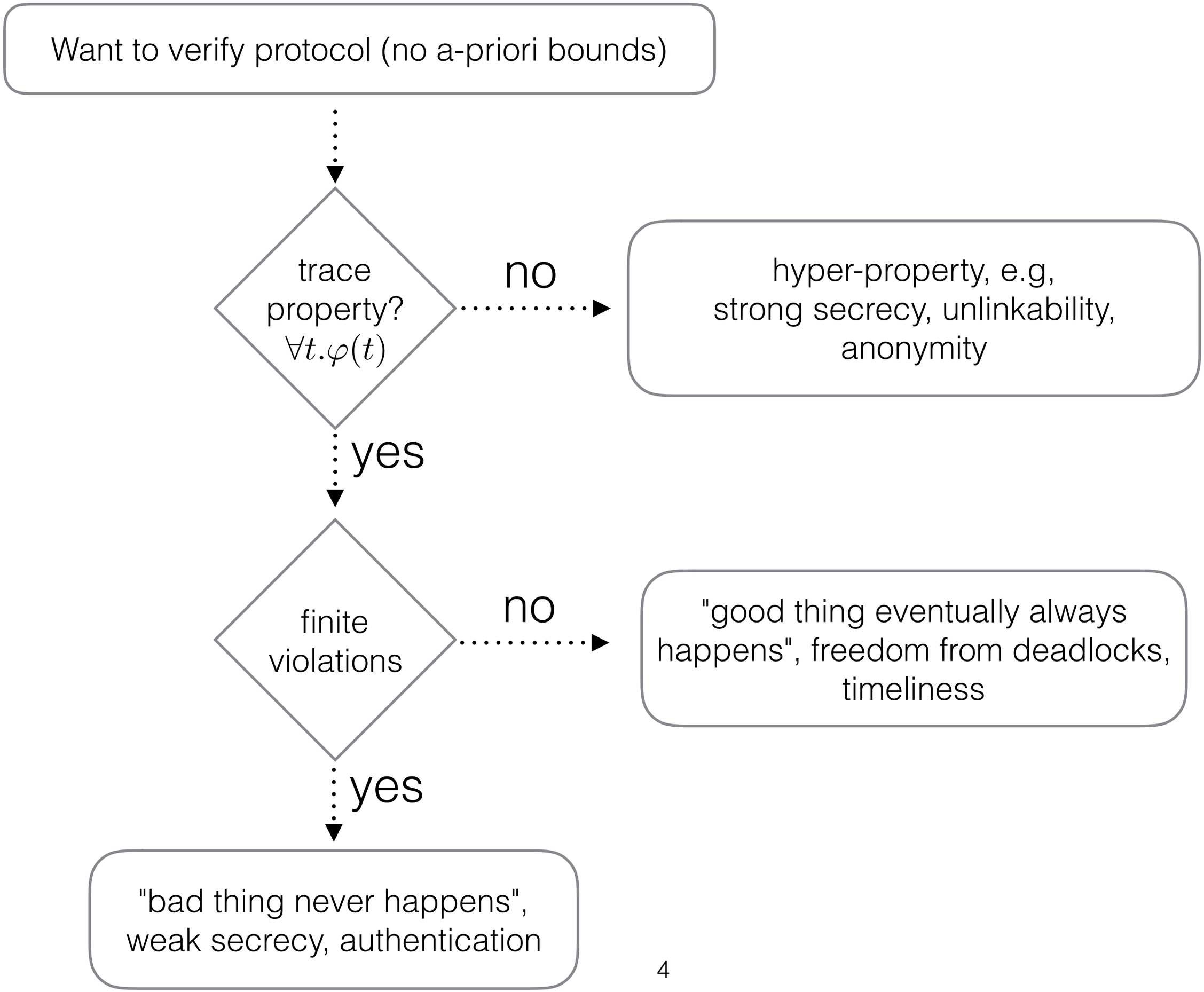
When do you need liveness?

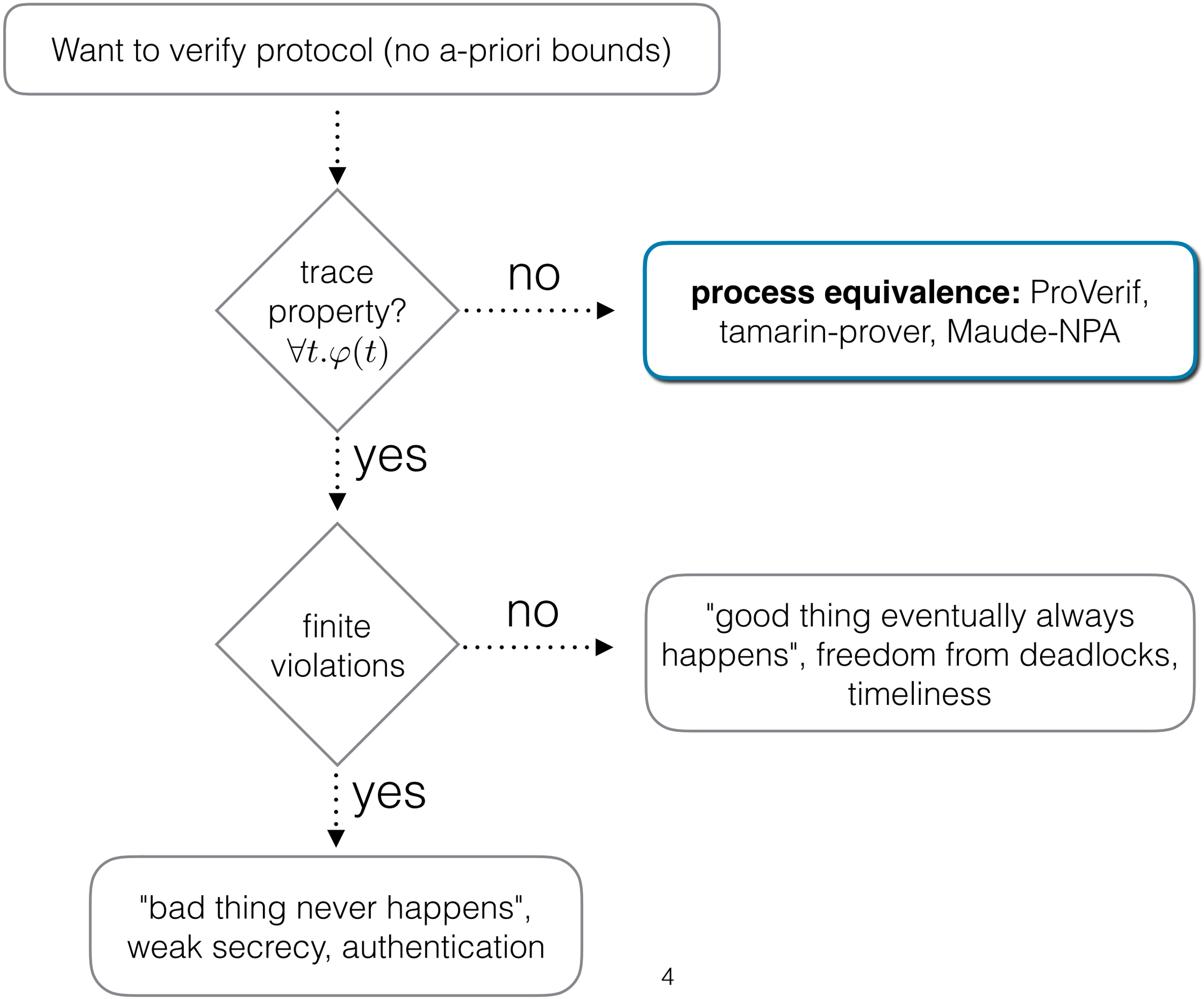
- when "no response" is dangerous, e.g., industrial control systems

- when recovery is a goal, e.g., self-healing crypto implementations

- when revocation or fairness are (sub)goals, e.g., accountability mechanisms




Want to verify protocol (no a-priori bounds)
trace property? $\forall t.\varphi(t)$
no
hyper-property, e.g, strong secrecy, unlinkability, anonymity
yes
finite violations
no
"good thing eventually always happens", freedom from deadlocks, timeliness
yes
"bad thing never happens", weak secrecy, authentication


Want to verify protocol (no a-priori bounds)
trace property? $\forall t.\varphi(t)$
no
process equivalence: ProVerif, tamarin-prover, Maude-NPA
yes
finite violations
no
"good thing eventually always happens", freedom from deadlocks, timeliness
yes
"bad thing never happens", weak secrecy, authentication

Want to verify protocol (no a-priori bounds)

trace property? $\forall t.\varphi(t)$

no

process equivalence: ProVerif, tamarin-prover, Maude-NPA

yes

finite violations

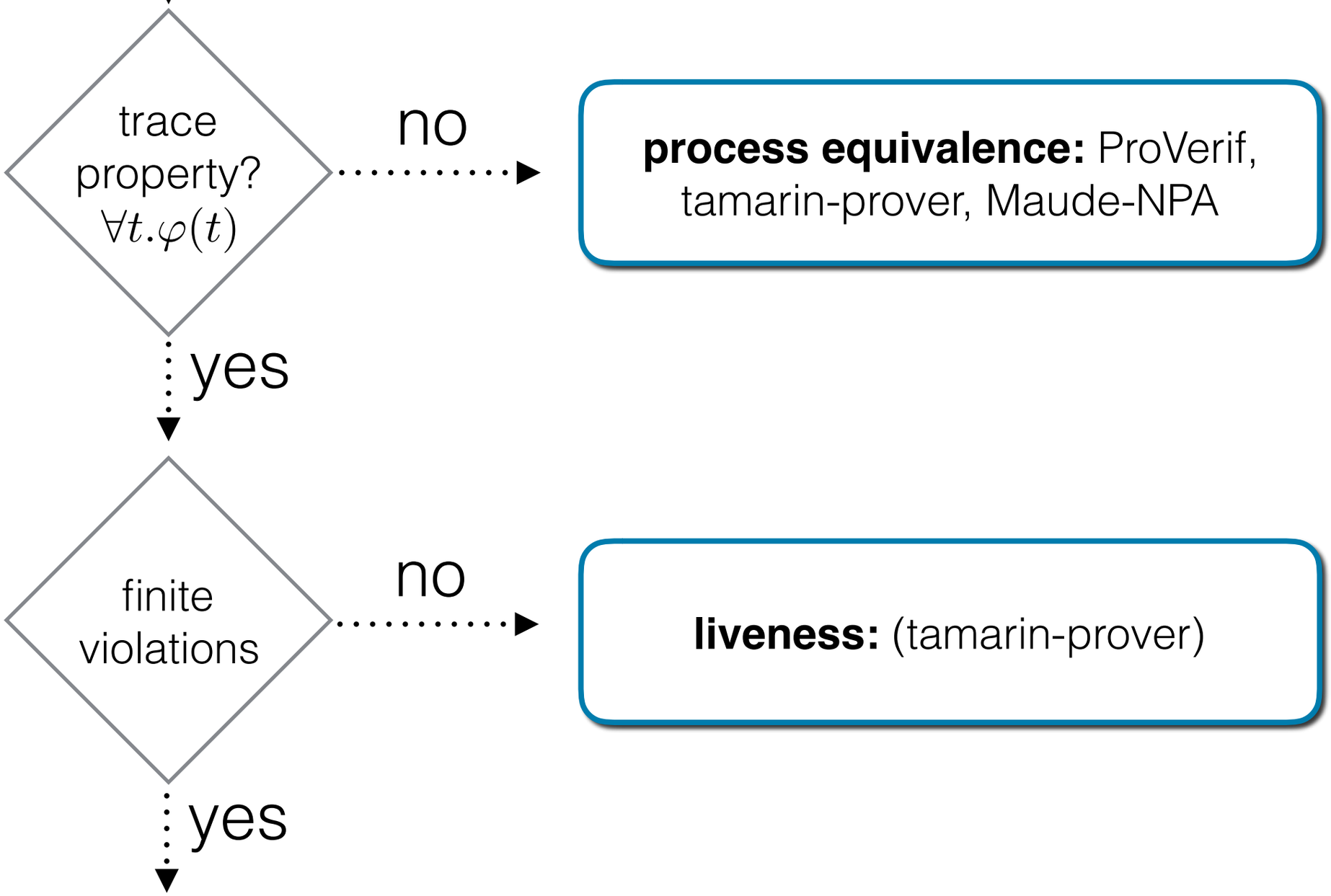
no

"good thing eventually always happens", freedom from deadlocks, timeliness

yes

safety: ProVerif, Maude-NPA, AVISPA, tamarin-prover, …

Want to verify protocol (no a-priori bounds)



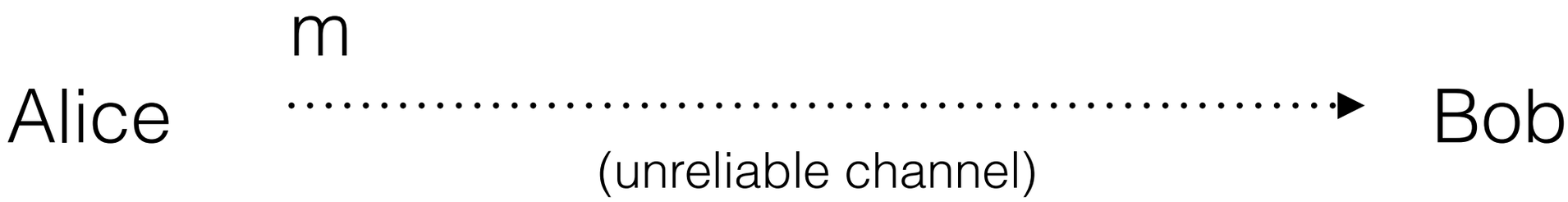
Obstacles

- Tamarin supports liveness properties, e.g.,

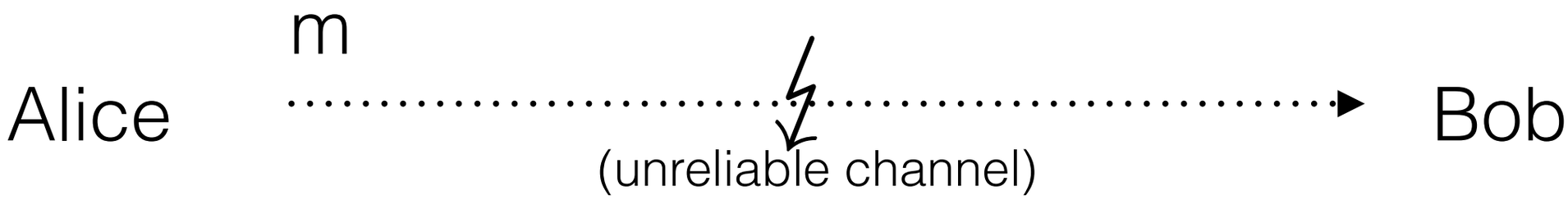
$$Send(A, m)@i \Longrightarrow Receive(B, m)@j$$

- however, we need assumptions…

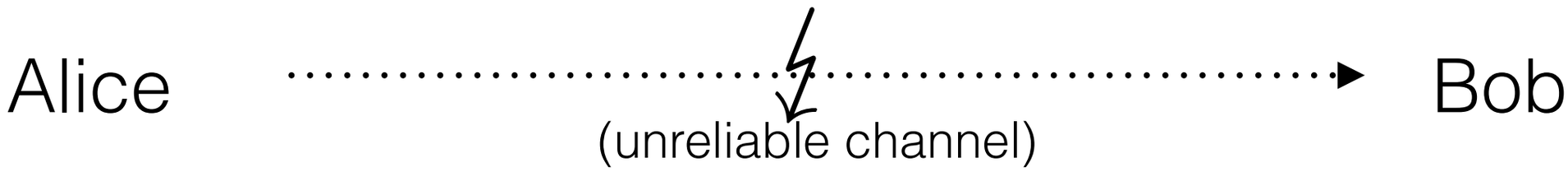
Assumptions



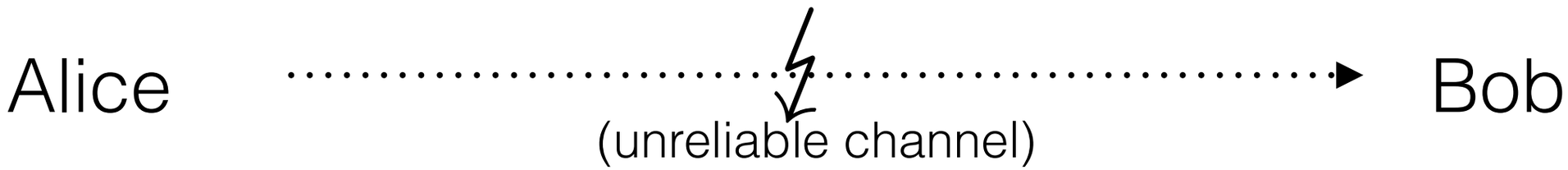
Assumptions



Assumptions

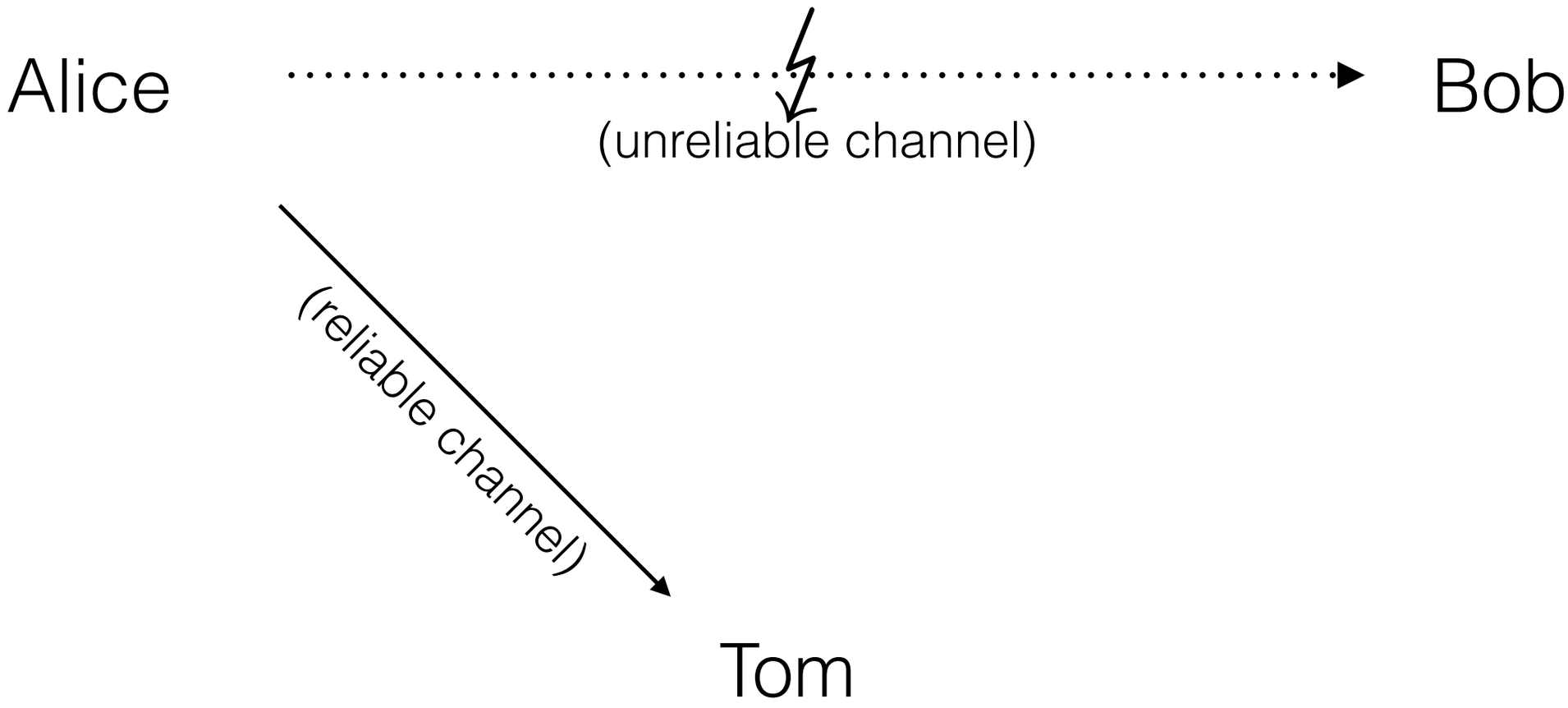


Assumptions

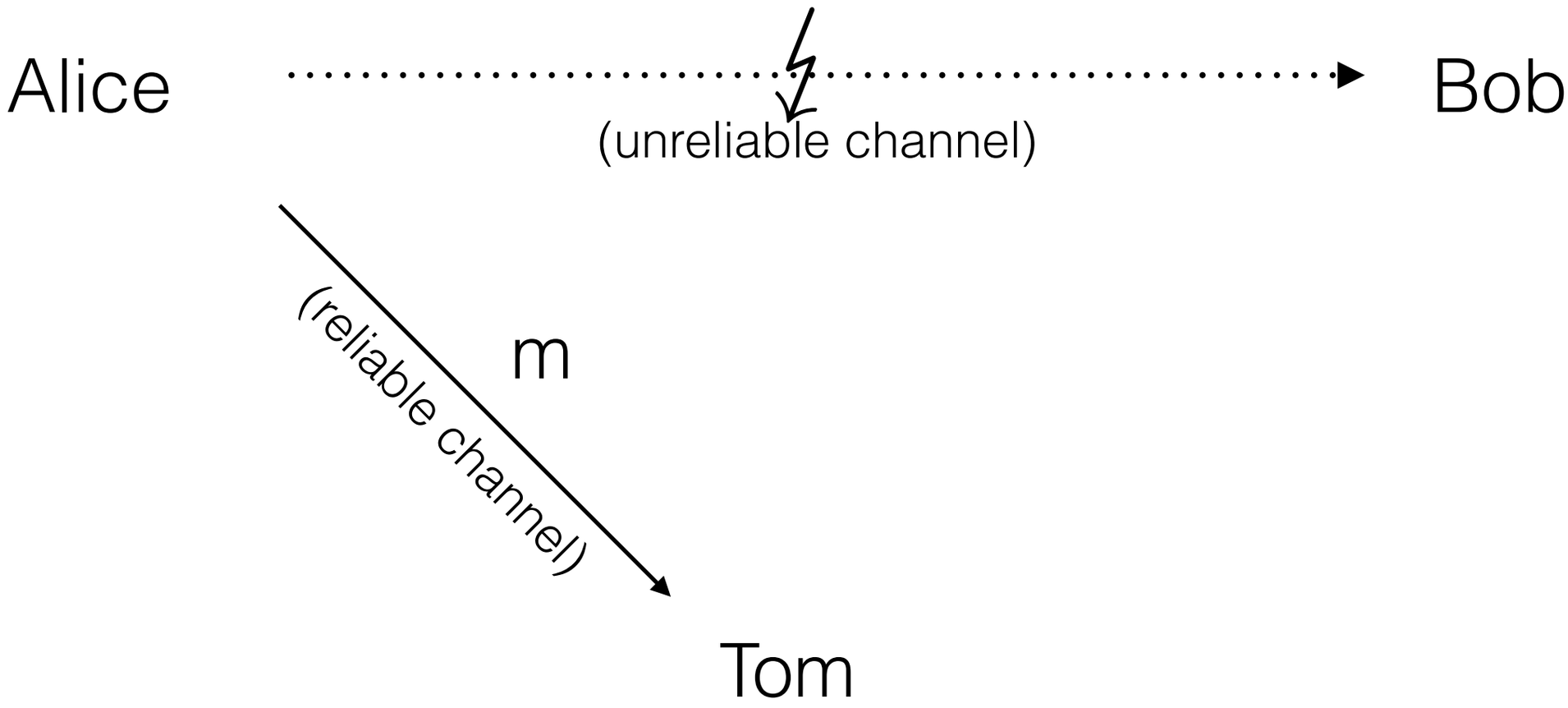


Tom

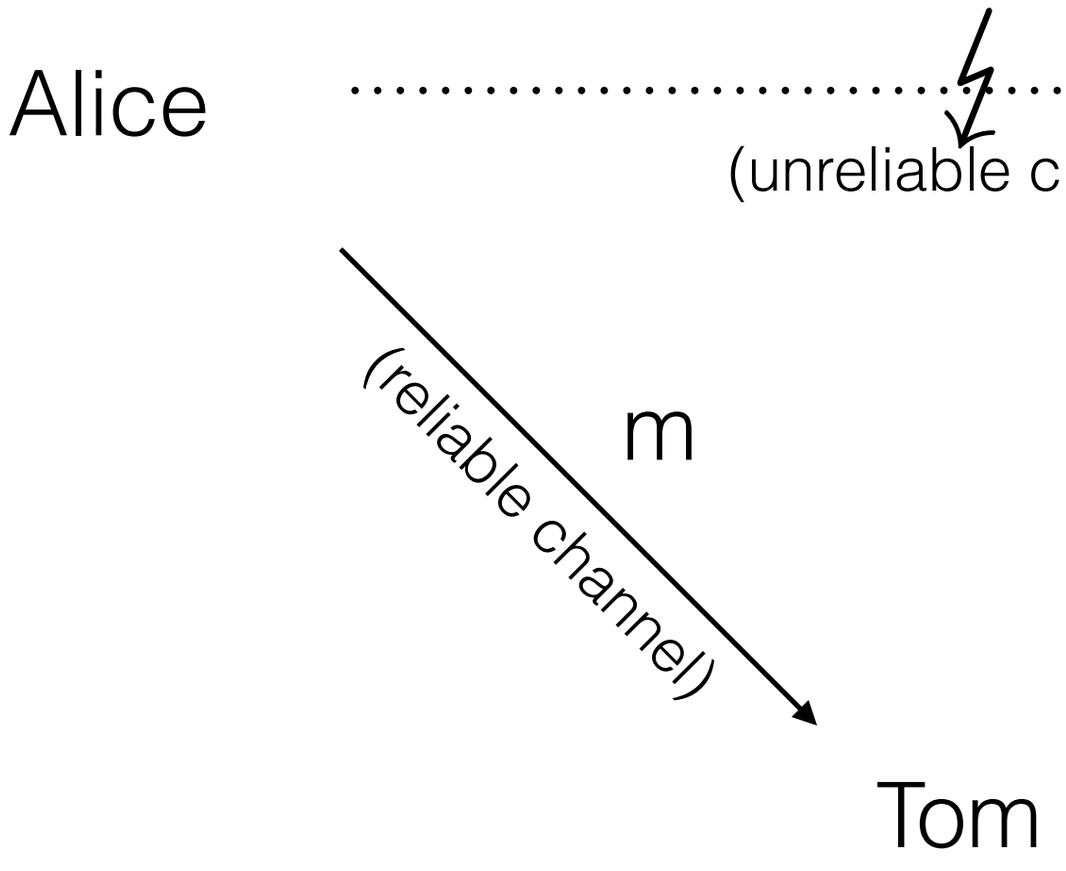
Assumptions



Assumptions

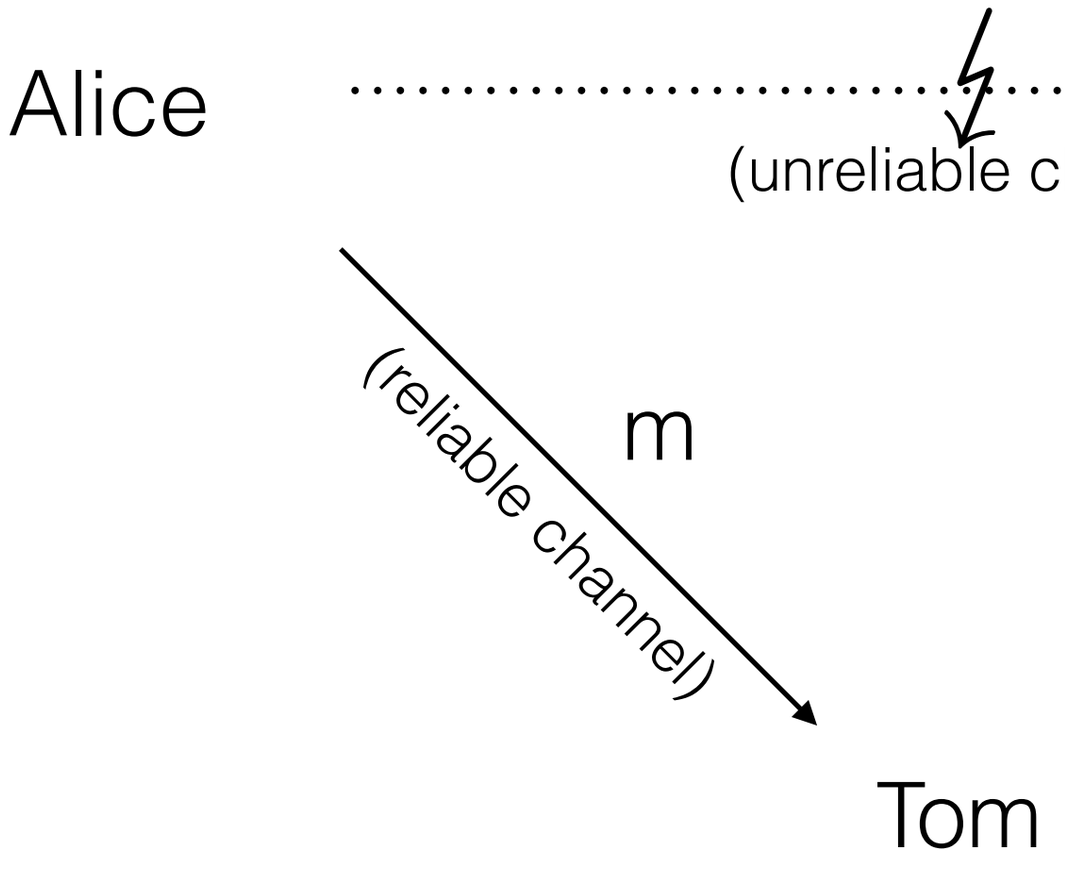


Assumptions



Alice needs to:

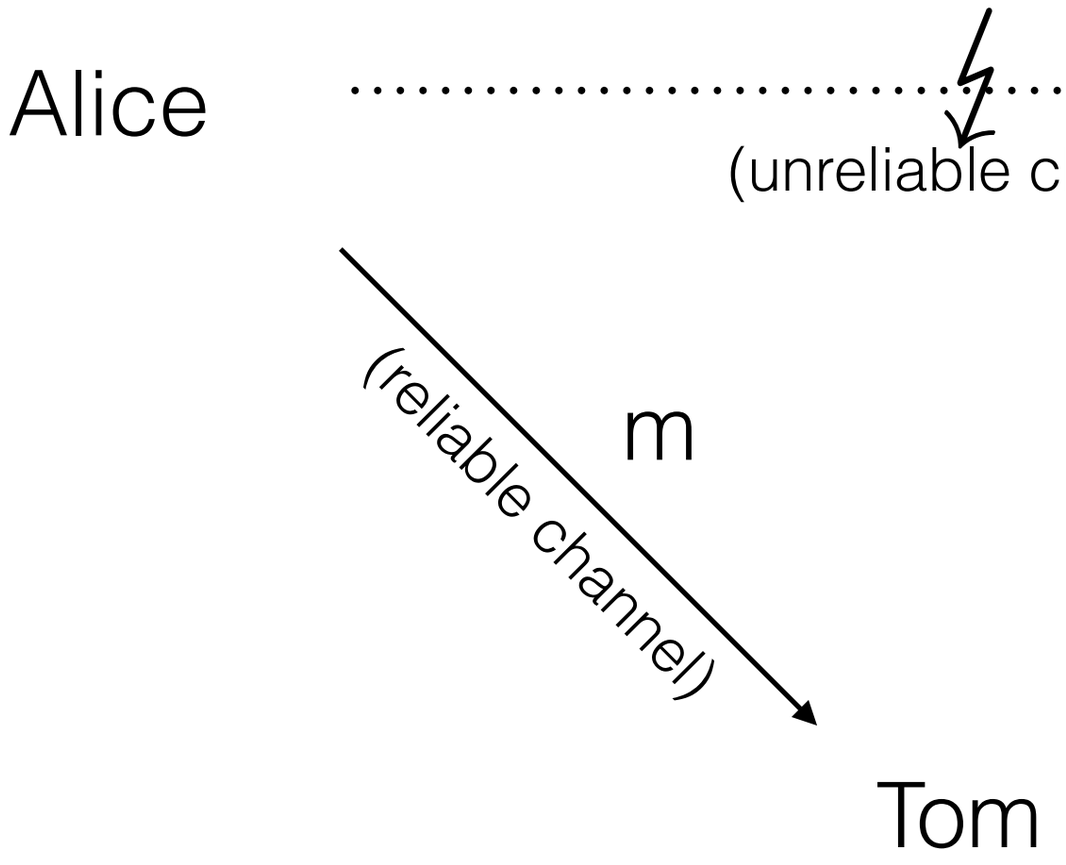
Assumptions



Alice needs to:

1. timeout after receiving no response from Bob,

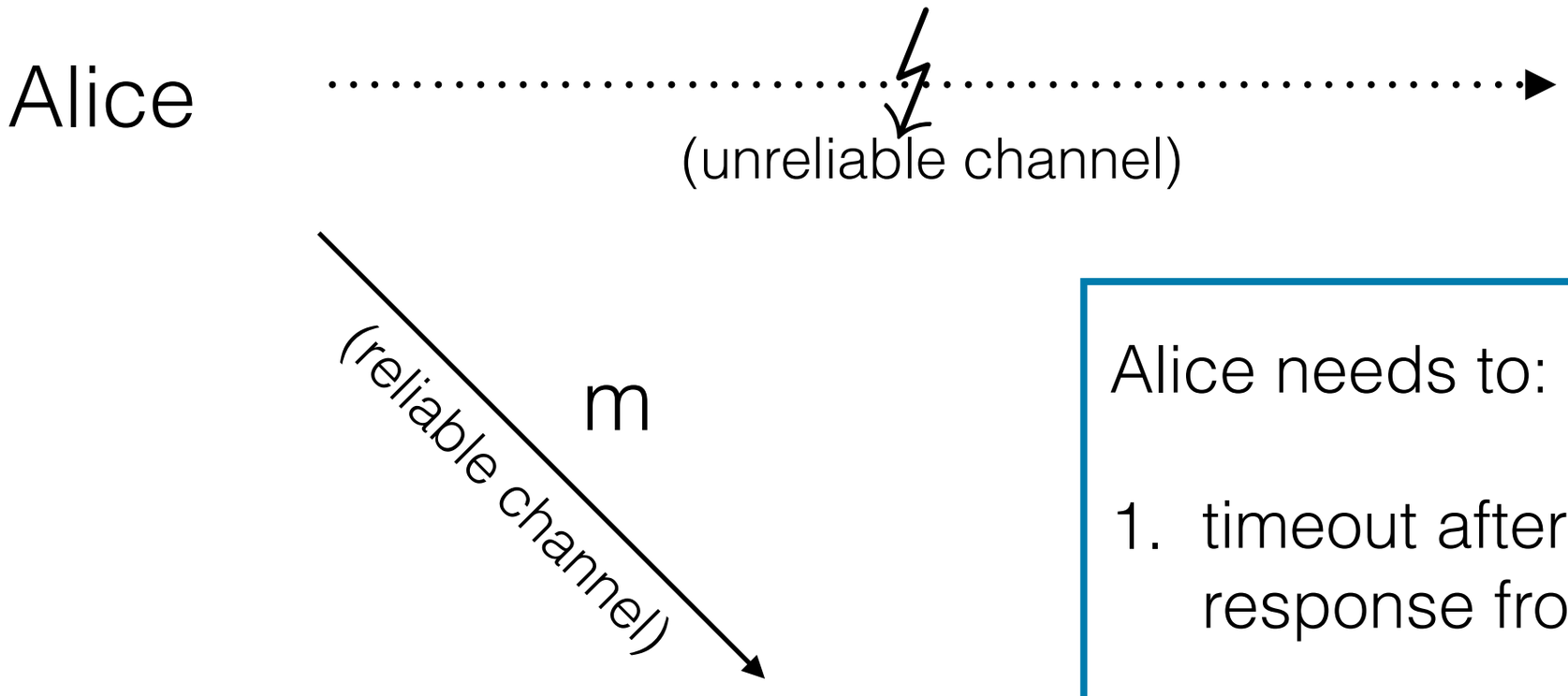
Assumptions



Alice needs to:

1. timeout after receiving no response from Bob,
2. then proceed to next step and

Assumptions



Alice needs to:

1. timeout after receiving no response from Bob,
2. then proceed to next step and
3. use a reliable channel to address Tom.

Assumptions (ctd)

- **Timeout:** (external) non-deterministic choice

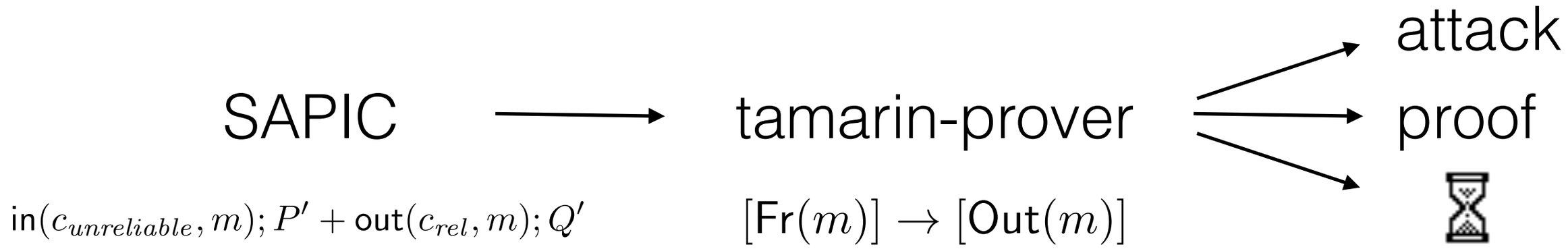
$$P + Q \to P' \text{ if } P \to P'$$

$$\mathsf{in}(c_{unreliable}, m); P' + \mathsf{out}(c_{rel}, m); Q' \xrightarrow{\mathsf{out}(\mathsf{c}_{rel},\mathsf{m})} Q'$$

- **local progress:** only 0, !P or in(m);P' are final states

- **reliable channels:** no undelivered messages in final state

Toolchain



- tamarin's multiset rewrite calculus is backend
- SAPIC: stateful applied-pi like calculus
- provides notion of locality, along with other extras
- completeness + soundness throughout (helping lemmas)

Contributions

- first toolchain for liveness in unbounded model
- characterisation of local progress
- heuristics for tamarin-prover (see next slide)

- analysis of:
  - Secure Conversation Protocol of OPA united architecture
  - ASW contract signing protocol (attack on original and Shmatikov/Mitchell's fix, fairness/timeliness for fixed version)
  - GJM contract signing protocol (attack, fairness/timeliness for fix)

Verification time

	ASW		ASW (mod.)		GJM		GJM (mod.)	
property	type	time	type	time	type	time	type	time
timeliness (A)	✓	1:40min	✓	1:38min	✓	0:46min	✓	6:08min
timeliness (B)	∞	—	✓	37:34min	✓	12:49min	✓	34:49h
fairness (A)	✗	8:34min	✗	31:06min	✗	2:22min	✓	14:11min
			✓	0:40h				
fairness (B)			✓	14:05h			✓	43:52h *

(* additional helping lemma verified in 2:38min)

Conclusion

- industrial setting and accountability become more important, so liveness becomes more important

Conclusion

- industrial setting and accountability become more important, so liveness becomes more important
- tool support so far limited to finite model checking, now high-level protocol calculus and mature protocol verifier for unbounded setting

Conclusion

- industrial setting and accountability become more important, so liveness becomes more important

- tool support so far limited to finite model checking, now high-level protocol calculus and mature protocol verifier for unbounded setting

- automation surprisingly good, however, computation power helps a lot