Rational Broadcast Protocols against Timid Adversaries

Keigo Yamashita and Kenji Yasunaga

Tokyo Institute of Technology

GameSec 2023@Avignon, France
Oct 19, 2023

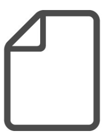
(Digital) Signature

Digital data for verifying the authenticity of messages

(Digital) Signature

Digital data for verifying the authenticity of messages

sk_A 

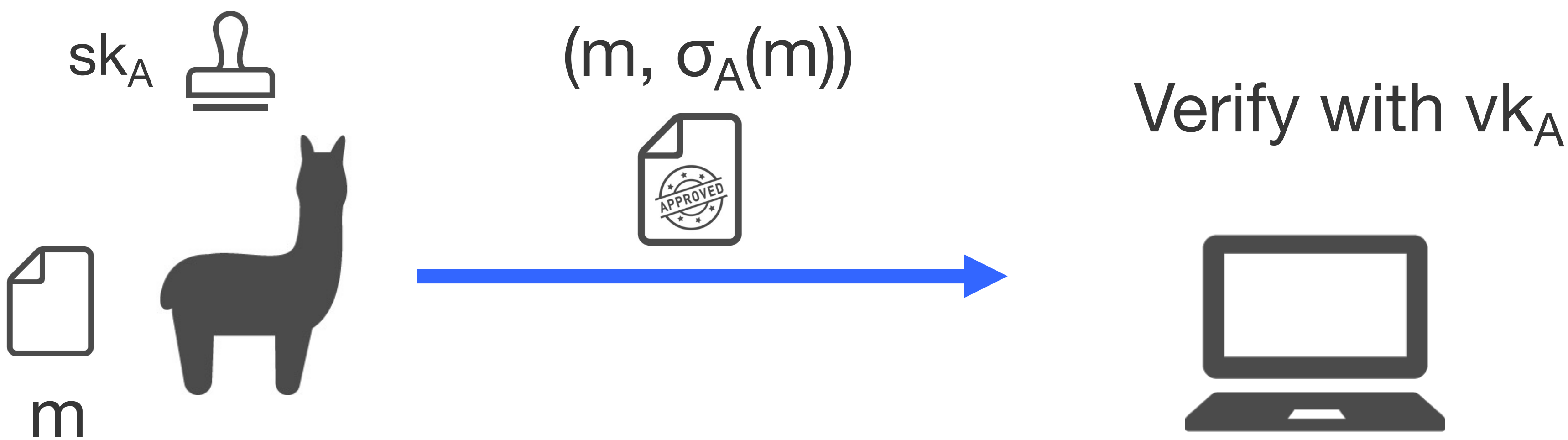
 

m

Verify with vk_A

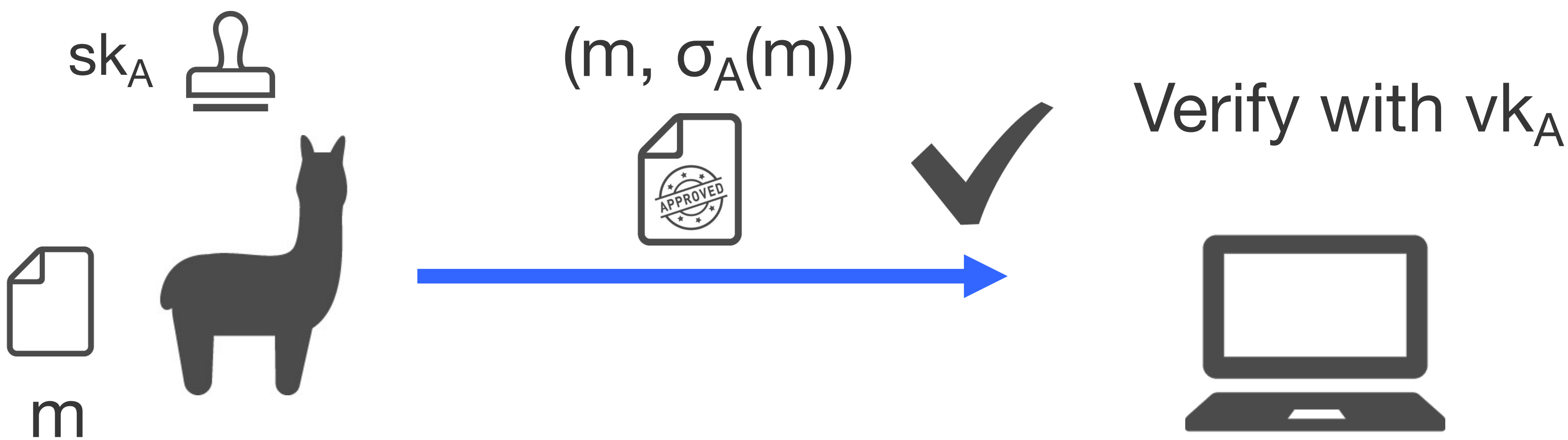
(Digital) Signature

Digital data for verifying the authenticity of messages



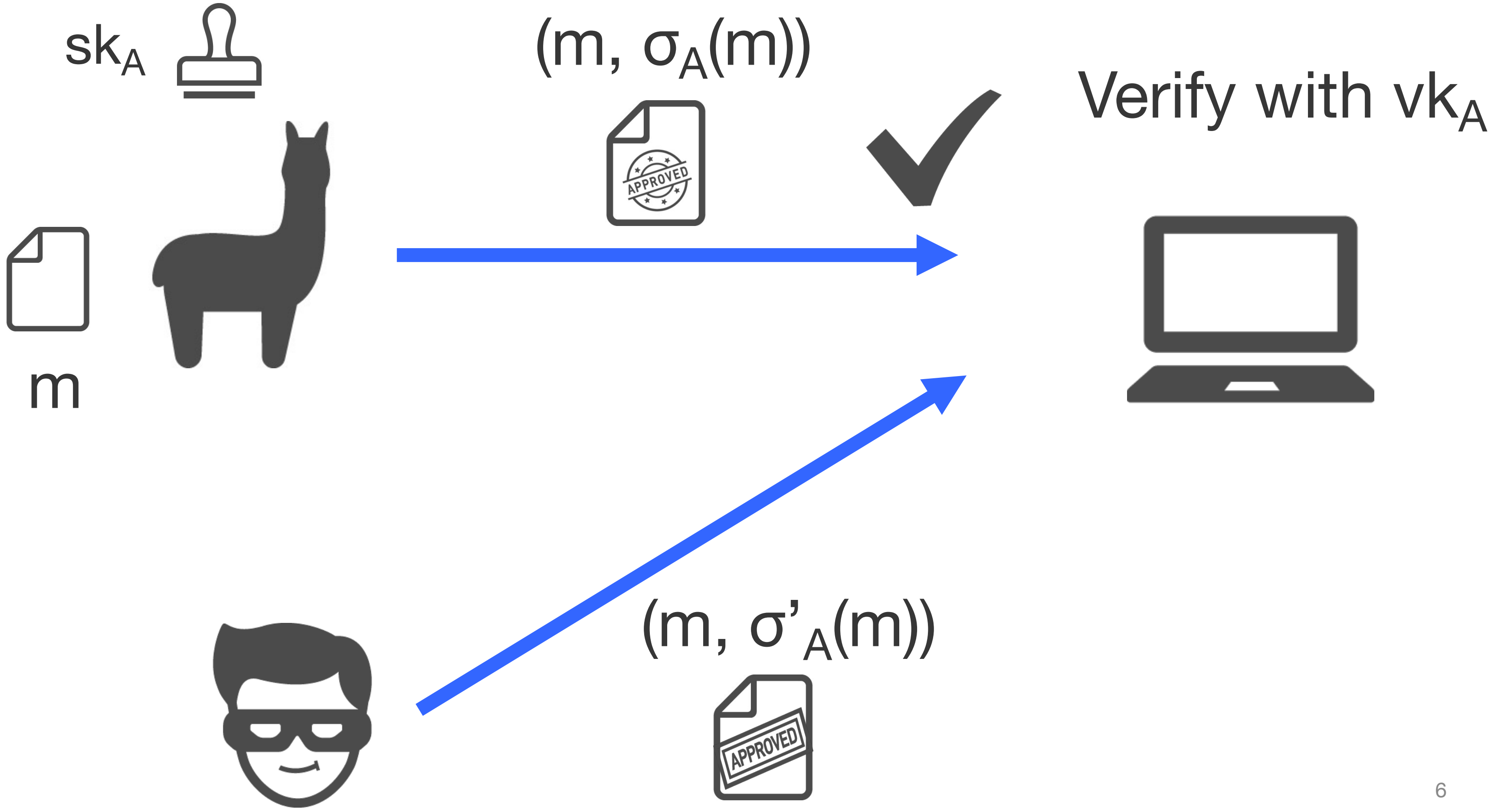
(Digital) Signature

Digital data for verifying the authenticity of messages



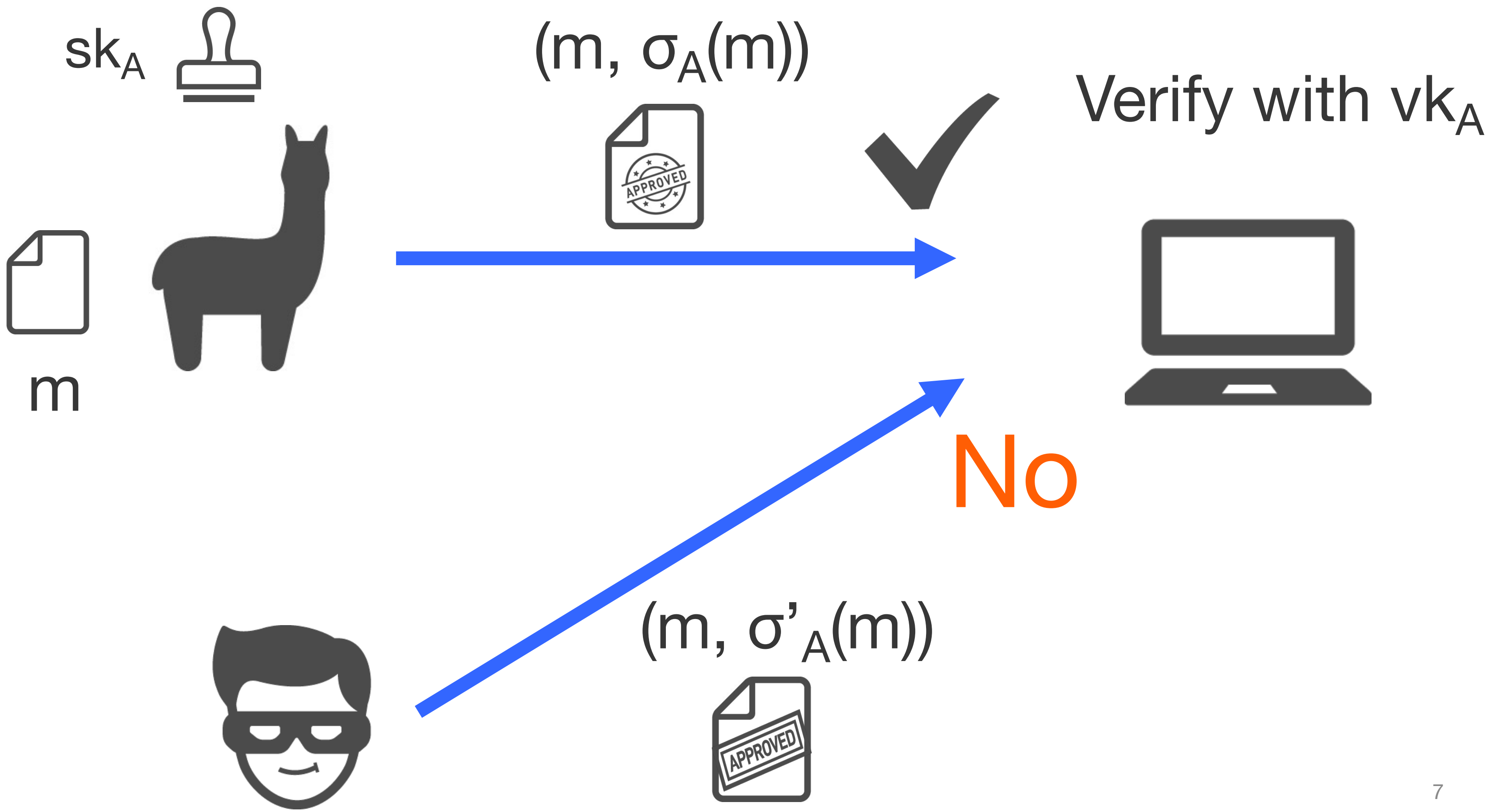
(Digital) Signature

Digital data for verifying the authenticity of messages



(Digital) Signature

Digital data for verifying the authenticity of messages

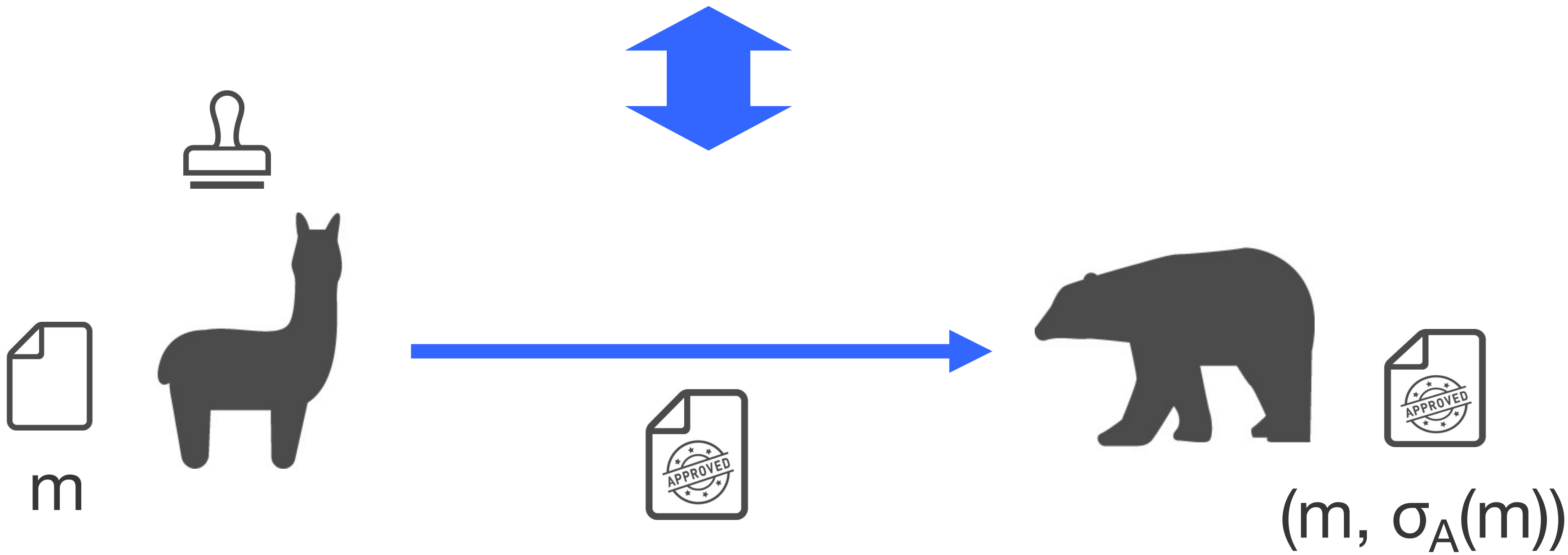


Role of Signature

B has a signature $(m, \sigma_A(m))$ of A

Role of Signature

B has a signature $(m, \sigma_A(m))$ of A



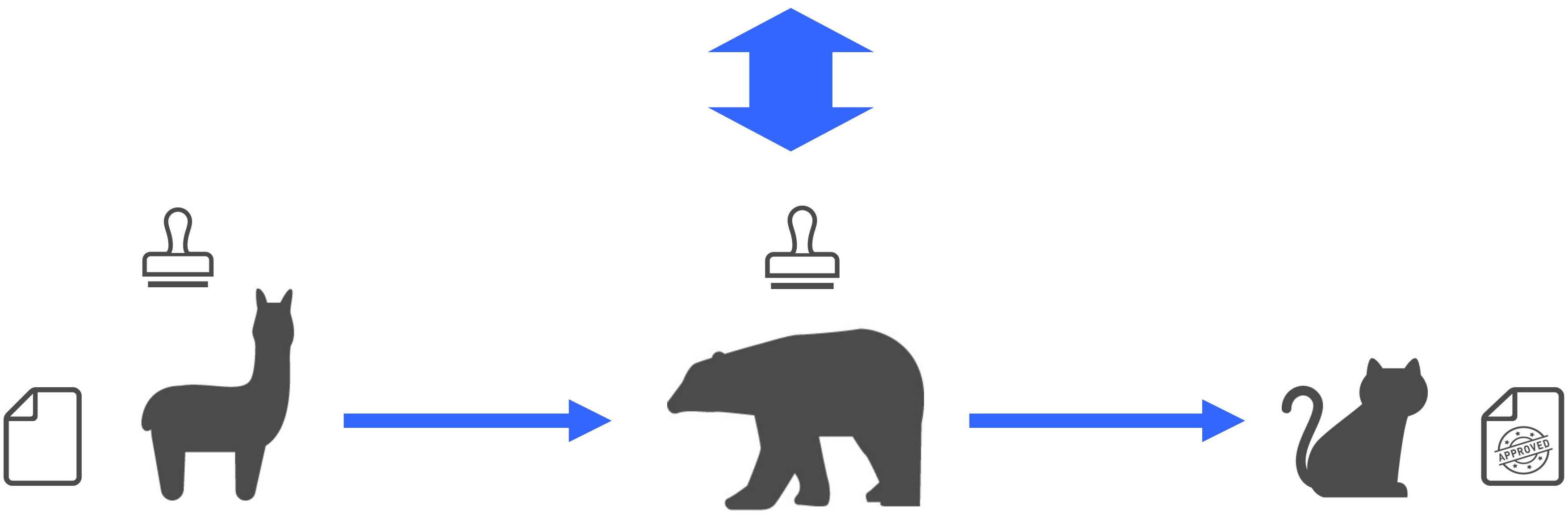
B has a proof that A sent m

Countersignature

C has a countersignature (m, $\sigma_B(\sigma_A(m))$)

Countersignature

C has a countersignature $(m, \sigma_B(\sigma_A(m)))$



m

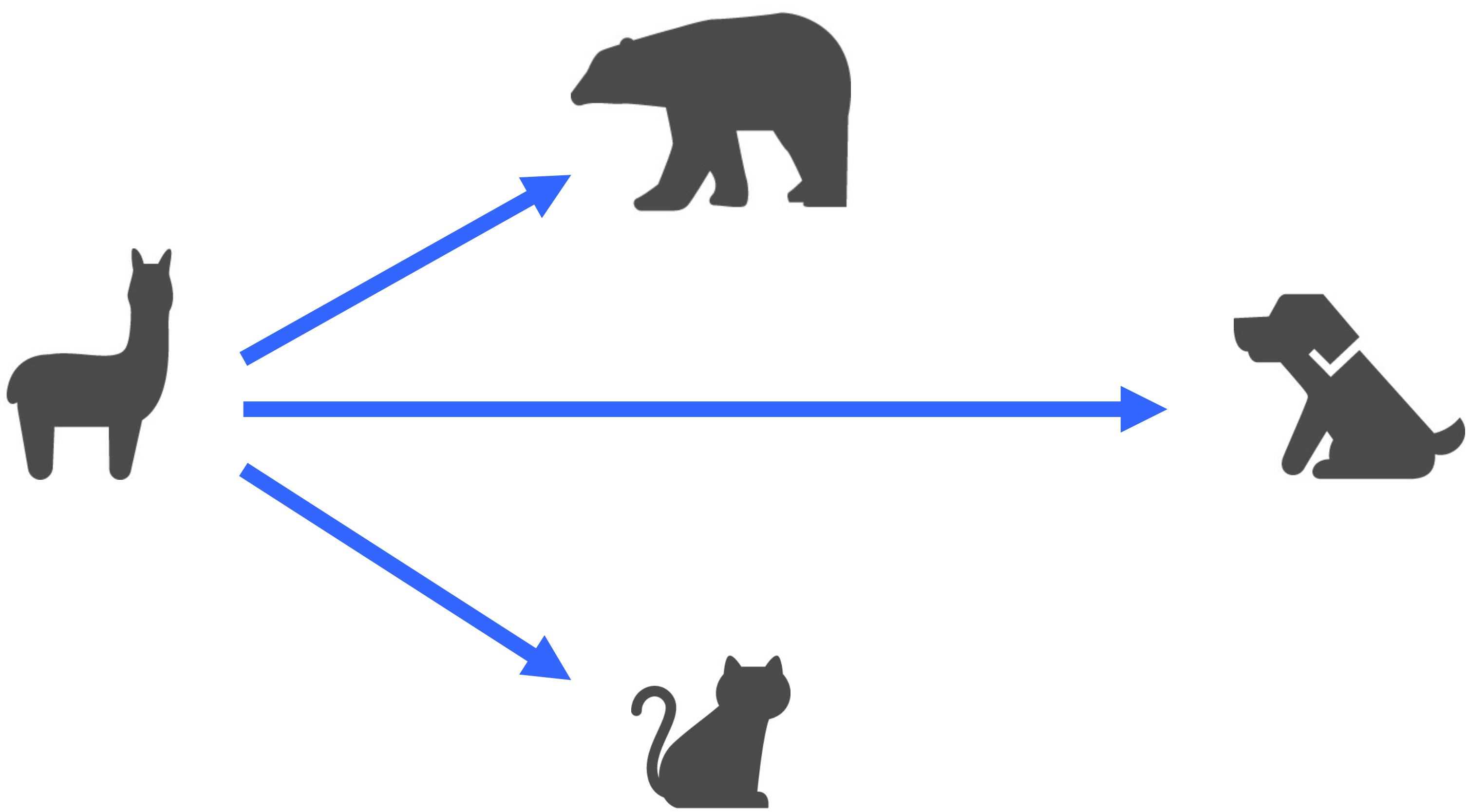
$(m, \sigma_B(\sigma_A(m)))$

C has a proof that B knows that A sent m

Broadcast Protocols

A sender sends the “same” message to all parties
even if the sender is malicious

- Building blocks for blockchains/multiparty computation



Broadcast Protocol (Setting & Requirements)

Broadcast Protocol (Setting & Requirements)

Setting

- A set $[n] = \{1, \ldots, n\}$ of n parties on secure P2P network
- Adversary can corrupt $\leq t$ parties
- Synchronous communication ($\exists$ rounds)
- PKI (Signature) is available (Authenticated setting)

Broadcast Protocol (Setting & Requirements)

Setting

- A set $[n] = \{1, \ldots, n\}$ of n parties on secure P2P network
- Adversary can corrupt $\leq t$ parties
- Synchronous communication ($\exists$ rounds)
- PKI (Signature) is available (Authenticated setting)

Requirements

- Validity: If a sender $s \in [n]$ with input m is honest (= not corrupted), all honest parties output m
- Agreement: All honest parties output the same value

Previous & Our Results on Authenticated Broadcast

References	Resilience	# Rounds	Adversary	Results
Dolev-Strong (1983)	$t < n$	$t+1$	Malicious	∃deterministic BC
Dolev-Strong (1983)	$t < n$	t	Malicious	No deterministic BC
Katz-Koo (2006)	$t < n/2$	29	Malicious	∃randomized BC
Garay et al. (2007)	$t < n/2+k$	$O(k^2)$	Malicious	∃randomized BC
Garay et al. (2007)	$t < n$	$o(2n/(n-t))$	Malicious	No randomized BC
Micali-Vaikuntanathan (2017)	$t < n/2$	$2\lambda+3$ w.p. $1-2^{-\lambda}$	Malicious	∃randomized BC
Abraham et al. (2019)	$t < n/2$	10	Malicious	∃randomized BC
This Work	$t < n$	5	Rational	∃deterministic BC

Our protocol runs in $t+5$ rounds for malicious adversaries

Rational adversary tries to maximize utility

Rational adversary tries to maximize utility

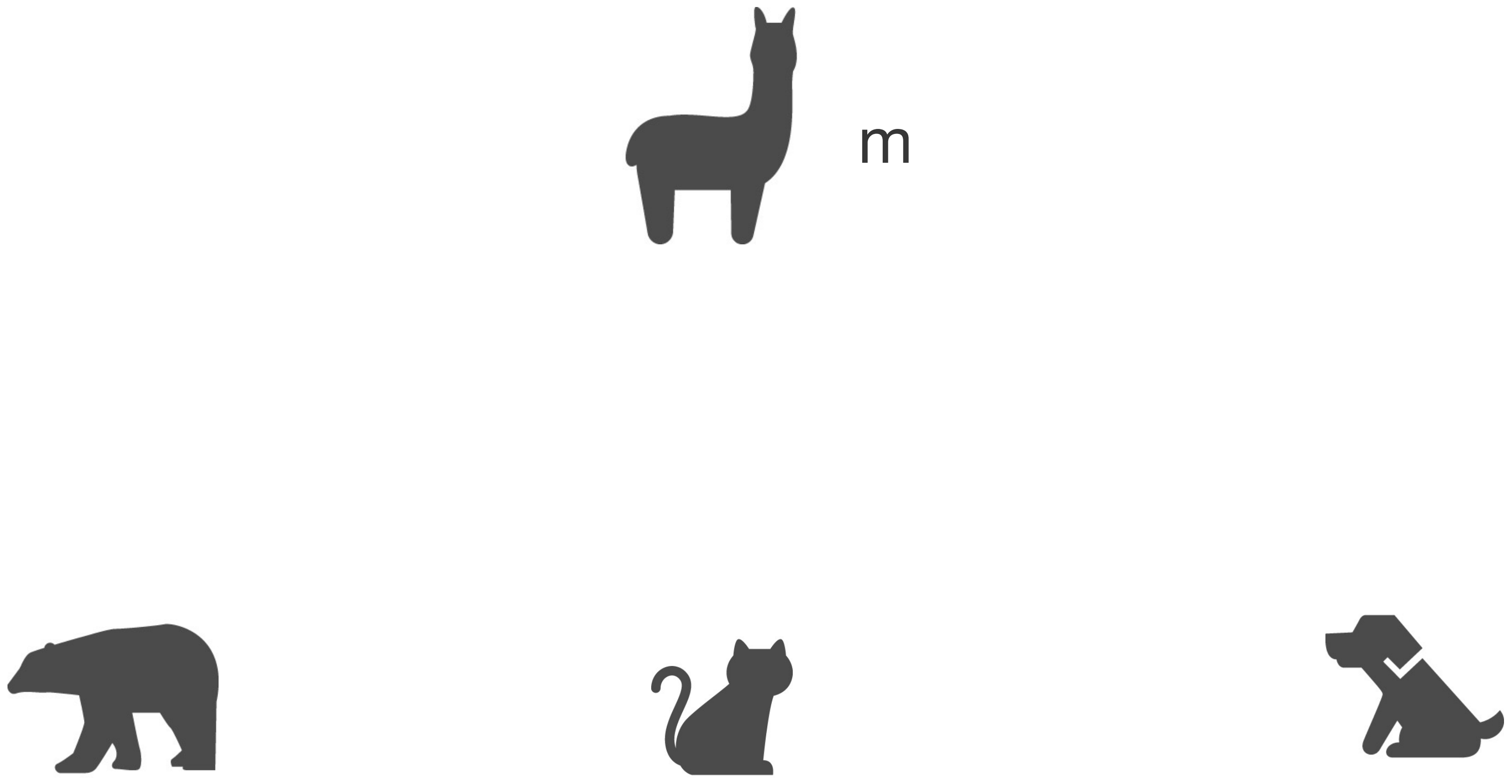
Timid adversary prefers to attack the protocol without being detected

Our Protocol

Our Protocol

<u>Round 1:</u>

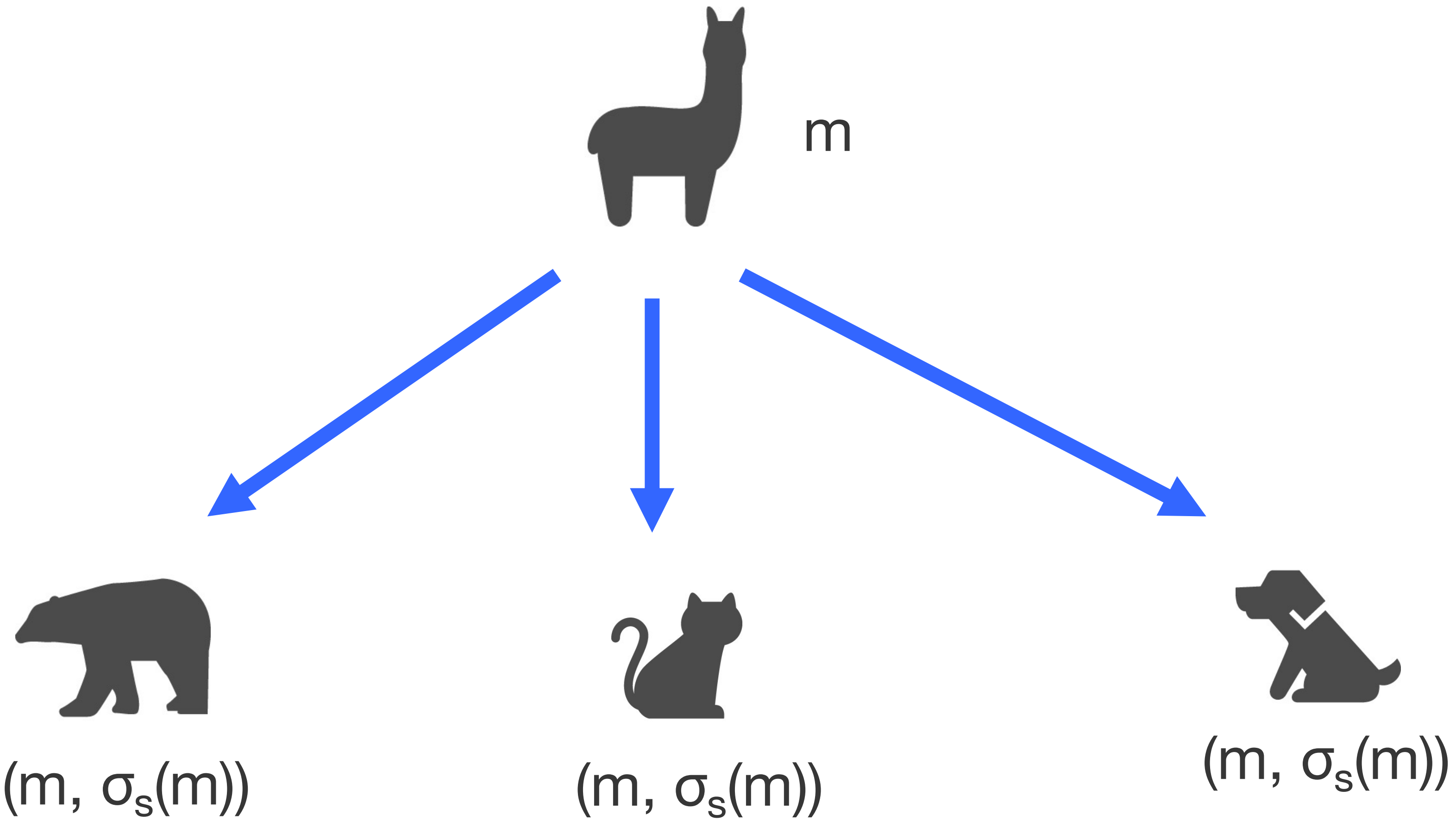
Sender s ∈ [n] sends (m, σ_s(m)) to all parties



Our Protocol

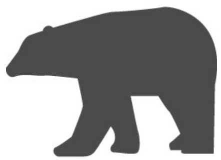
Round 1:

Sender $s \in [n]$ sends $(m, \sigma_s(m))$ to all parties



Round 2:

Each $i \in [n]$ sends countersig $(m, \sigma_i(\sigma_s(m)))$ to all parties



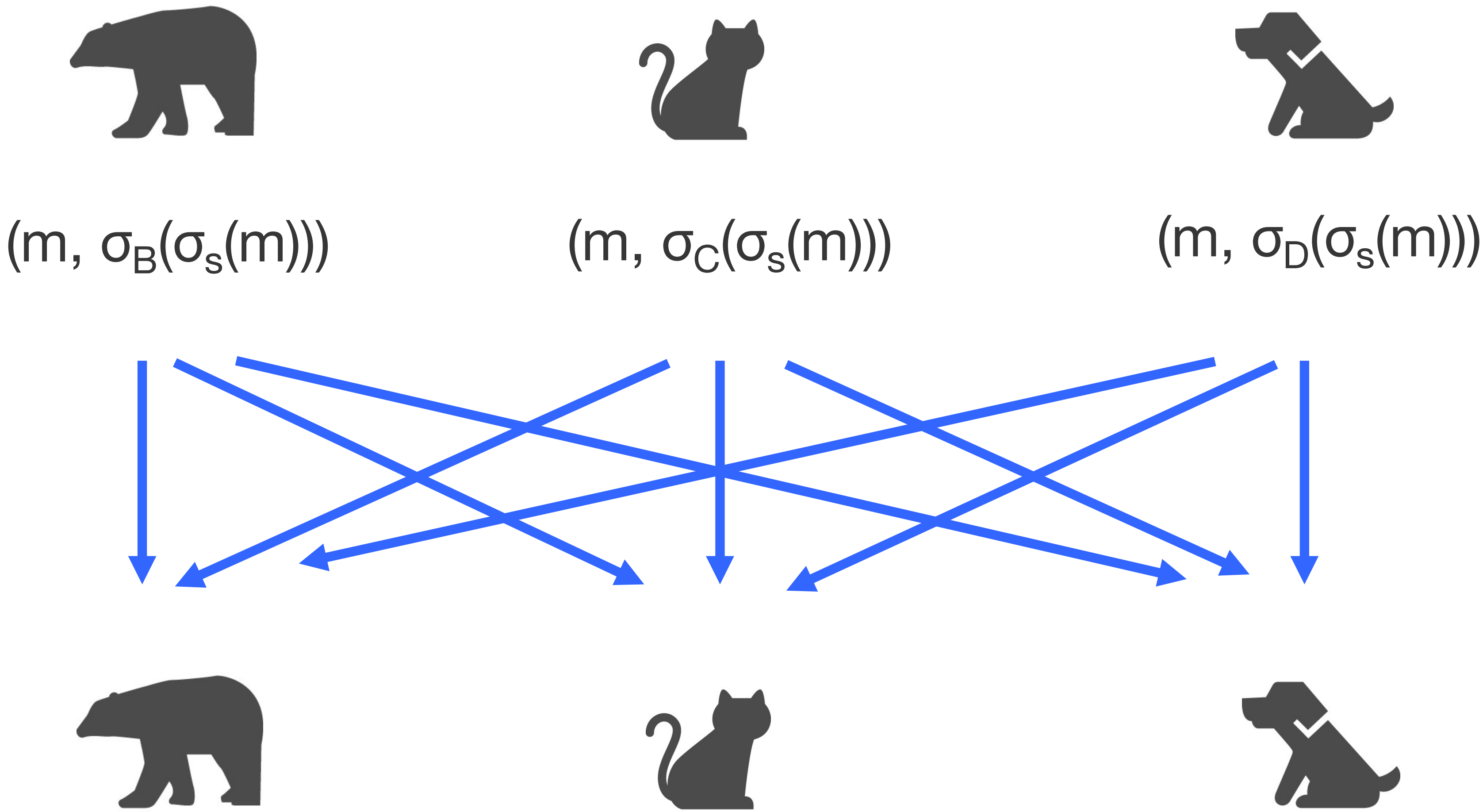
$(m, \sigma_B(\sigma_s(m)))$

$(m, \sigma_C(\sigma_s(m)))$

$(m, \sigma_D(\sigma_s(m)))$

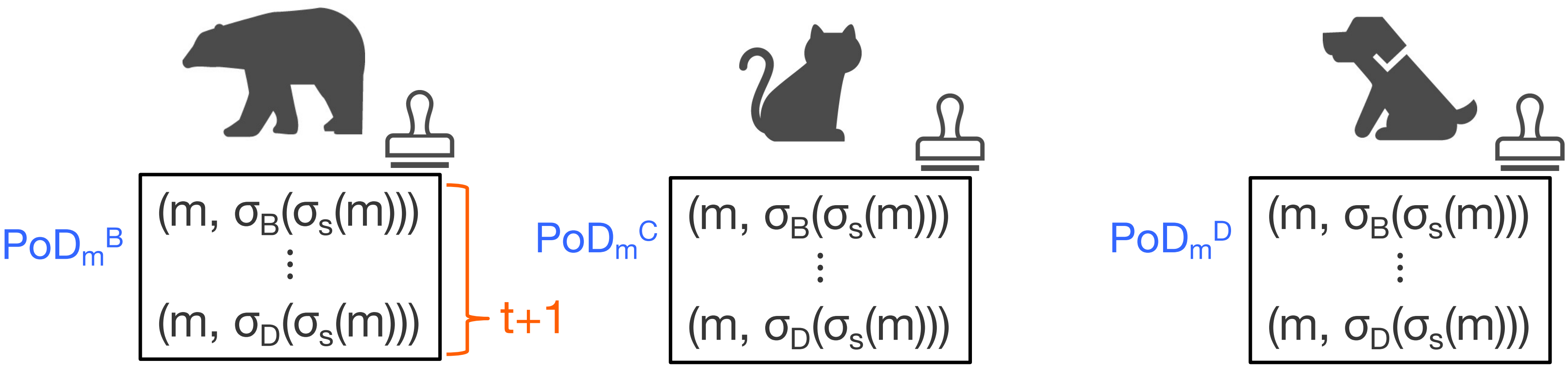
Round 2:

Each $i \in [n]$ sends countersig $(m, \sigma_i(\sigma_s(m)))$ to all parties



Round 3:

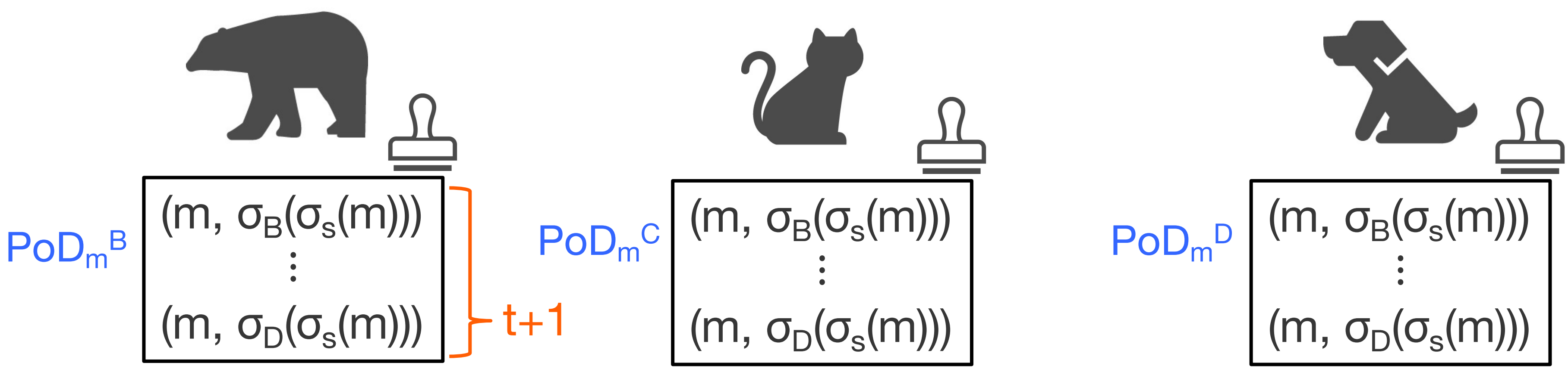
Each $i \in [n]$ collects t+1 valid countersigs to generate a signed "proof of dissemination" (PoD_m^i) and sends it to all parties



Round 3:

∃honest party's countersig, which was sent to all parties

Each $i \in [n]$ collects t+1 valid countersigs to generate a signed "proof of dissemination" (PoD_m^i) and sends it to all parties

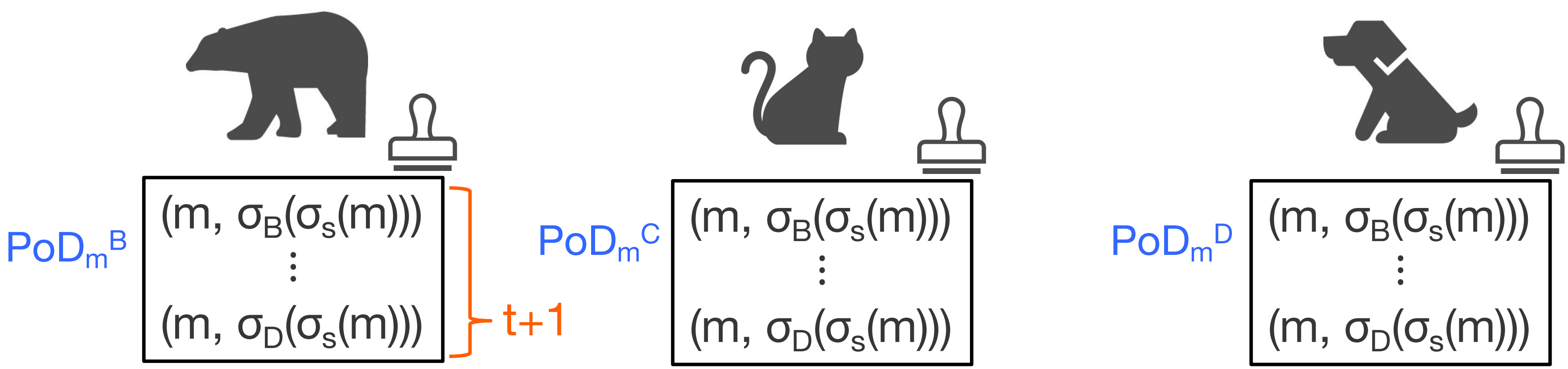


Round 3:

∃honest party's countersig, which was sent to all parties

Each $i \in [n]$ collects t+1 valid countersigs to generate a signed "proof of dissemination" (PoD_m^i) and sends it to all parties

PoD_m^i = i knows that everyone got a countersig for m

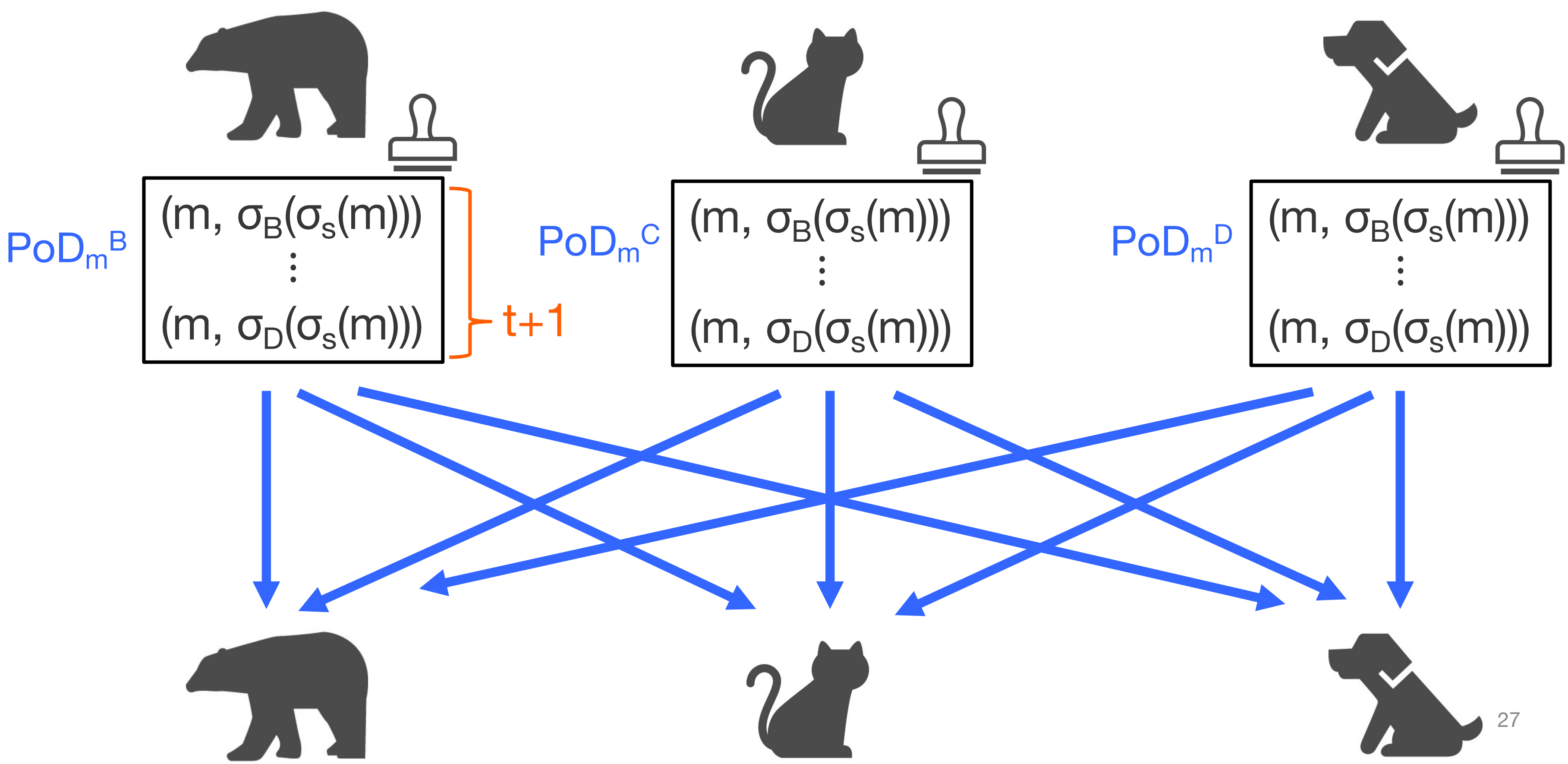


Round 3:

∃honest party's countersig, which was sent to all parties

Each i ∈ [n] collects t+1 valid countersigs to generate a signed "proof of dissemination" (PoD_m^i) and sends it to all parties

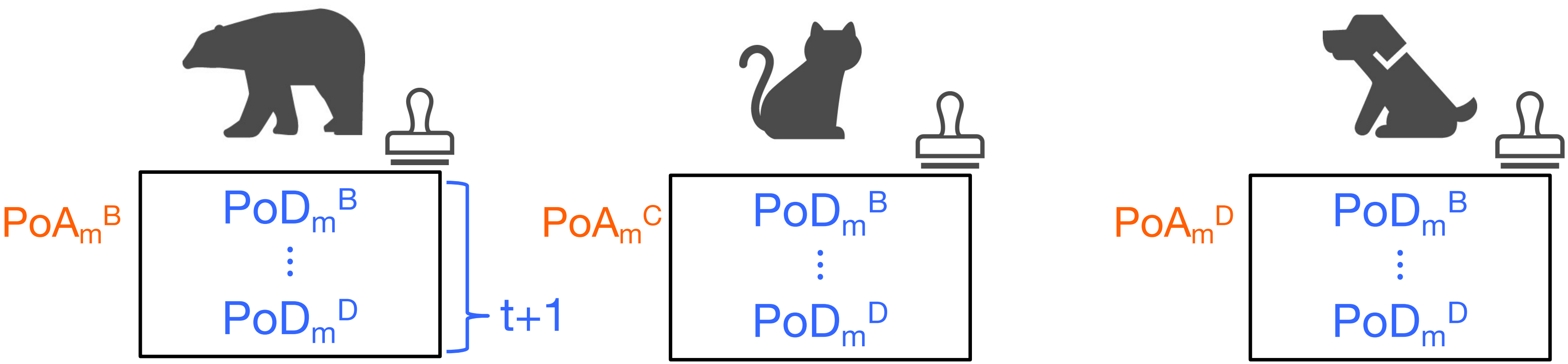
PoD_m^i = i knows that everyone got a countersig for m



Round 4:

Each i ∈ [n] collects t+1 valid PoD_m^j to generate a signed "proof of agreement" (PoA_m^i) and sends it via Dolev-Strong protocol

(If i sees valid PoD_m & $PoD_{m'}$ for distinct m & m', i does nothing)



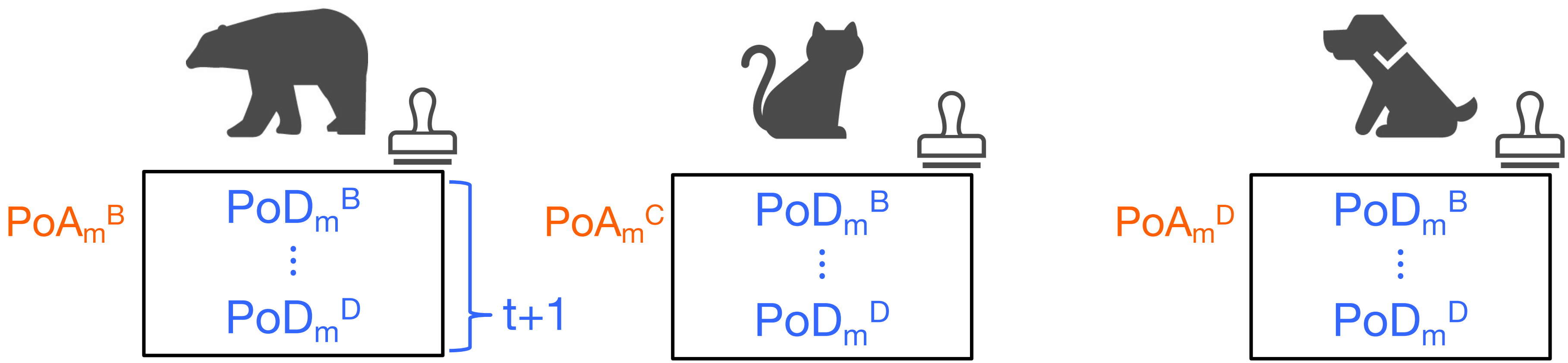
Round 4:

Each $i \in [n]$ collects $t+1$ valid PoD_m^j to generate a signed "proof of agreement" (PoA_m^i) and sends it via Dolev-Strong protocol

(If i sees valid PoD_m & $PoD_{m'}$ for distinct m & m', i does nothing)

> PoA_m^i = i knows that everyone knows that everyone got a countersig for m

Another party may not know PoA_m



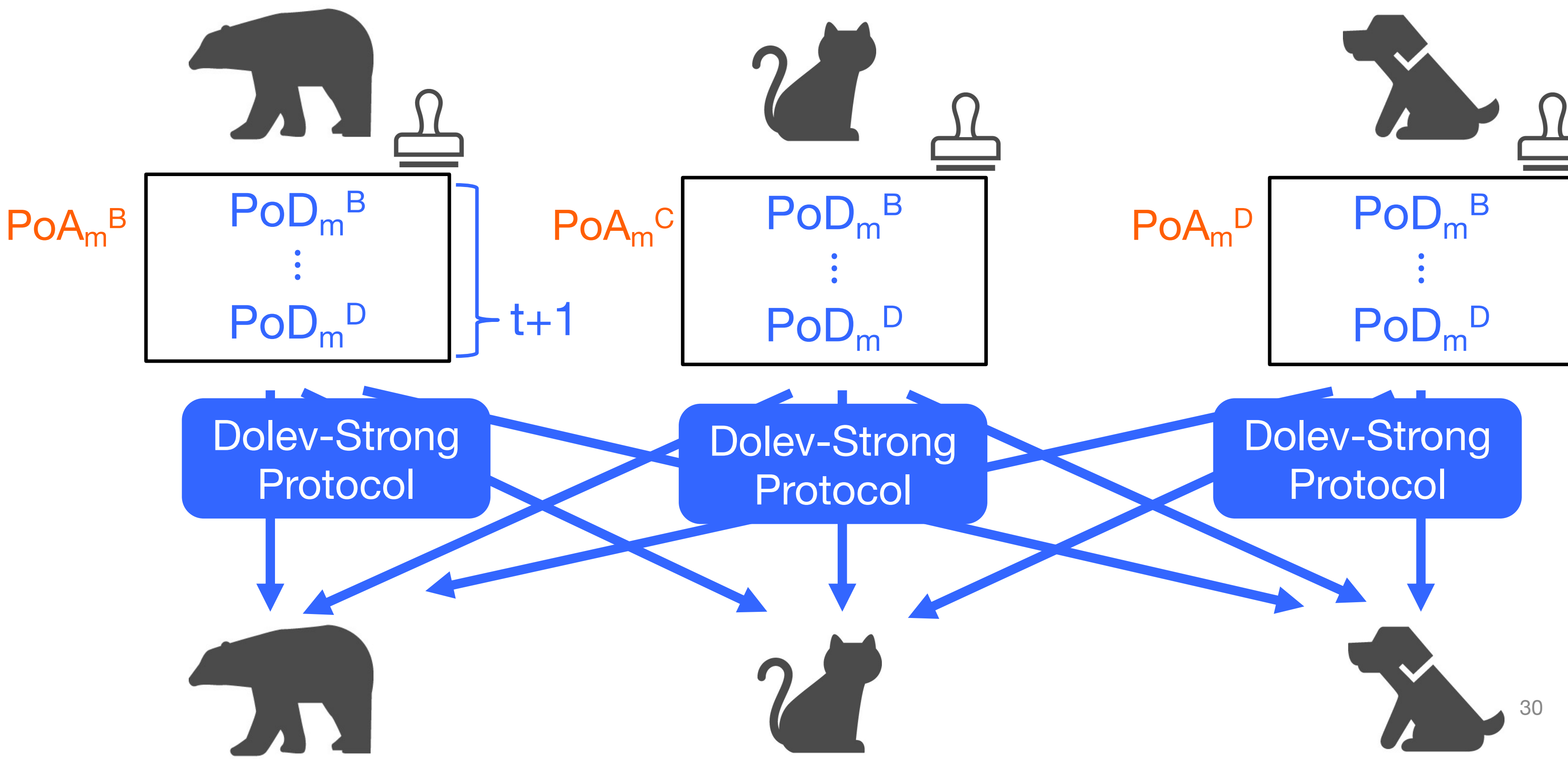
Round 4:

Each $i \in [n]$ collects t+1 valid PoD_m^j to generate a signed "proof of agreement" (PoA_m^i) and sends it via Dolev-Strong protocol

(If i sees valid PoD_m & $PoD_{m'}$ for distinct m & m', i does nothing)

PoA_m^i = i knows that everyone knows that everyone got a countersig for m

Another party may not know PoA_m



Round 5:

If i ∈ [n] collects t+1 valid PoA_m^j (= “proof of termination” (PoT)), outputs m and halts.

Otherwise, i continues DS protocol.

Round 5:

If $i \in [n]$ collects $t+1$ valid PoA_m^j (= "proof of termination" (PoT)), outputs m and halts.
Otherwise, i continues DS protocol.

Round t+5:

If DS protocol outputs valid PoA_m, i outputs m.
Otherwise i outputs $\perp$ and sends "DETECT s" (s is cheating)

Round 5:

If $i \in [n]$ collects $t+1$ valid PoA_m^j (= "proof of termination" (PoT)), outputs m and halts.
Otherwise, i continues DS protocol.

Round t+5:

If DS protocol outputs valid PoA_m, i outputs m.
Otherwise i outputs $\perp$ and sends "DETECT s" (s is cheating)

Key Observations:

No party can obtain PoA_m & $PoA_{m'}$ for $m \neq m'$ simultaneously
(If so, every honest party sees PoD_m & $PoD_{m'}$ ➔ No PoA exists)

1. Honest party i output $m \neq \perp$ ➔ i obtained PoA_m
2. Honest party i output $\perp$ ➔ Every honest party failed to get PoA

Theorem

For any adversary corrupting t ($< n$) parties, our protocol satisfies

- weak validity
- agreement

The protocol finishes in round 5 for timid adversaries

- If finishes in round $t+5$ (output $\perp$), the sender's cheating is detected.

Requirements:

- Weak validity: If a sender $s \in [n]$ with input m is honest, all honest parties output m or $\perp$
- Agreement: All honest parties output the same value

Proof Overview

1. When violating weak validity:

   Sender s with input m is honest & Honest party i output $m'(\neq m)$

   → i got $PoA_{m'}$, but s never generates a signature for m'

   → Contradiction

2. When violating agreement with $(out_i, out_j) = (m, m'(\neq m))$:

   → i got PoA_m & j got $PoA_{m'}$ **→ Contradicting the observations**

3. When violating agreement with $(out_i, out_j) = (m, \perp)$:

   "$out_i = m$" → Honest i got PoA_m

   "$out_j = \perp$" → Every honest party failed to get PoA

   Contradiction

Discussion (False Detection)

When $t \geq n/2$,
honest sender s may be falsely detected as a cheater

- If $t = n/2$ parties do nothing, valid PoD cannot be generated

  → Honest party outputs $\perp$ (and s is declared cheating)

When $t < n/2$,
honest sender s can never be detected as a cheater

Conclusions

Construct a 5-round deterministic broadcast protocol against timid adversaries for $t < n$

- Avoiding DS lower bound by rationality
- Round complexity is $t+5$ in the worst (malicious) case

Future Work

- Improve the round complexity
- Construct a protocol without false detection for $t \geq n/2$ (or prove its impossibility)
- Achieve (standard) validity for $t \geq n/2$

Conclusions

Construct a 5-round deterministic broadcast protocol against timid adversaries for $t < n$

- Avoiding DS lower bound by rationality
- Round complexity is $t+5$ in the worst (malicious) case

Future Work

- Improve the round complexity
- Construct a protocol without false detection for $t \geq n/2$ (or prove its impossibility)
- Achieve (standard) validity for $t \geq n/2$

Broadcast Game for protocol Π

1. Set incorrect = disagree = undetect = 0
2. Adversary A chooses sender $s \in [n]$, message m, corrupted parties $C \subseteq [n]$ with $|C| \leq t$
3. Run Π where s is the sender with message m and A controls parties in C
4. After running Π, each $i \in [n]$ outputs v_i. Let $H = [n] \setminus C$.

   If $s \in H$ & $\exists\ i \in H$ s.t. $v_i \notin \{m, \perp\}$, set incorrect = 1
   If $\exists i, j \in H$ s.t. $v_i \neq v_j$, set disagree = 1
   If no party sent "DETECT", set undetect = 1
5. Outcome is out = (incorrect, disagree, undetect)

Utility of Timid Adversary

- For two outcomes out = (incorr, disag, undet) and out' = (incorr', disag', undet'),

1. U(out) > U(out')
   if incorr > incorr', disag = disag', undet = undet'
2. U(out) > U(out')
   if incorr = incorr', disag > disag', undet = undet'
3. U(out) > U(out')
   if incorr = incorr', disag = disag', undet > undet'

- By definition,

$$U(1,1,1) > \max\{ U(0,1,1), U(1,0,1) \}$$
$$\geq \min\{ U(0,1,1), U(1,0,1) \} > U(0,0,1) > U(0,0,0)$$

Security of Rational Broadcast

Protocol Π is secure against rational t-adversaries with U

⇔

∃(harmless) adversary B controlling ≤ t parties s.t.

1. Security: Π satisfies validity and agreement for B
2. Nash equilibrium:
   For every A controlling ≤ t parties, u(A) ≤ u(B).
   - u(A) := E[U(out_A)] is the expected value of U(out) for A

Dolev-Strong Protocol

Round 1:

Sender $s \in [n]$ sends $(m, \sigma_s(m))$ to all parties

Round $r = 2, \ldots, t+1$:

Each party $i \in [n]$, on receiving $c = (c_1, c_2)$,
if c_2 is a $(r - 1)$-fold valid signature of distinct signers $\neq i$,
then sends $(c, \sigma_i(c))$ to all parties. (Once for each m)
Otherwise, i sends nothing.

The end of round t+1:

Let V be the set of values of $(t+1)$-fold valid signatures.
If $|V| = 1$, output the value in V. Otherwise, output $\perp$.

$(t+1)$-fold valid signature of m
= everyone got the proof that s sent m