Decentralized Business Networks: Security

Jonas Kastberg Hinrichsen
Aalborg University

Willard Rafnsson
IT University of Copenhagen

Yvonne Dittrich
IT University of Copenhagen

Kim Peiter Jørgensen
Tecminho, University of Minho

Ravi Prakash
FIDE Foundation

Decentralized Business Networks: Security

Why do we care about security in decentralized business networks?

Decentralized Business Networks: Security

Why do we care about security in decentralized business networks?

Networks

- Huge attack vector

Decentralized Business Networks: Security

Why do we care about security in decentralized business networks?

Networks

- Huge attack vector

Business

- High stakes

Decentralized Business Networks: Security

Why do we care about security in decentralized business networks?

Networks

- Huge attack vector

Business

- High stakes

Decentralized

- Non-trivial wrt. enforcement, accountability, etc.

Decentralized Business Networks: Security (Continued)

Two levels of attack surfaces

- **Infrastructure-level:** What happens outside the protocol
- **Communication-level:** What happens inside the protocol

Decentralized Business Networks: Security (Continued)

Two levels of attack surfaces

- **Infrastructure-level:** What happens outside the protocol
- **Communication-level:** What happens inside the protocol

Infrastructure-level security

- Threat Model: External attackers (MitM, replay, tampering, etc.)
- Policy: Confidentiality, integrity, availability
- Mechanism: Encryption, signatures, nonces
- Assurance: Vetted off-the-shelf solutions

Decentralized Business Networks: Security (Continued)

Two levels of attack surfaces

- **Infrastructure-level:** What happens outside the protocol
- **Communication-level:** What happens inside the protocol

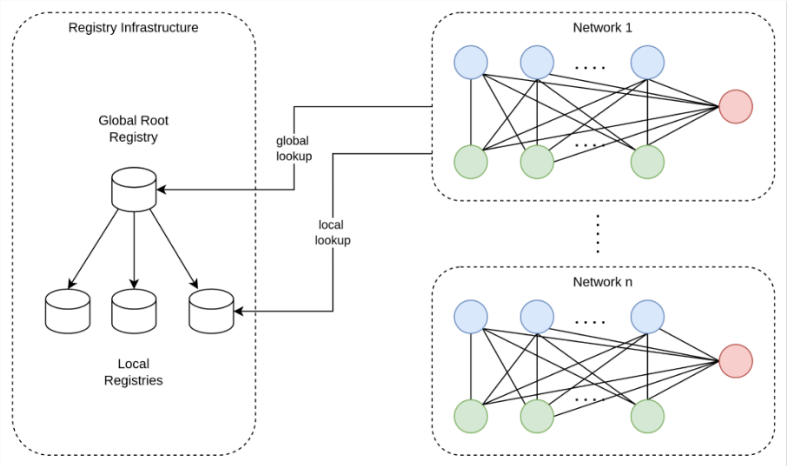
Infrastructure-level security

- Threat Model: External attackers (MitM, replay, tampering, etc.)
- Policy: Confidentiality, integrity, availability
- Mechanism: Encryption, signatures, nonces
- Assurance: Vetted off-the-shelf solutions

Communication-level security

- Threat Model: Internal (protocol-conforming) attackers
- Policy: Privacy, availablity, fraud-avoidance, etc.
- Mechanism: Protocol specification, properties, and conformance
- Assurance: Certification, sampling, traffic analysis

Beckn Architecture

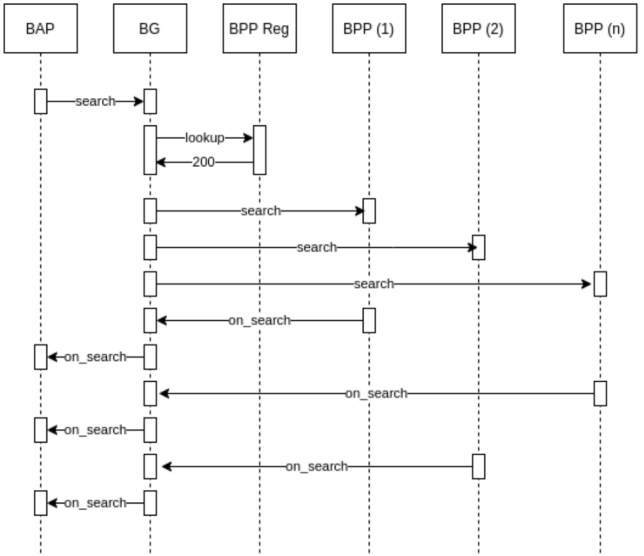


Blue: Buyers (BAPs), Green: Sellers (BPPs), Red: Gateways (BGs)

Beckn API

Stage	Method	Description	Returns
Discovery	`search`	declare intent	catalog
Order	`select`	draft order	quote
	`init`	shipping/billing info	payment terms
	`confirm`	confirm order	acknowledgment
Fulfillment	`status`	get order	order details
	`track`	track order	tracking details
	`update`	update order	acknowledgment
	`cancel`	cancel order	acknowledgment
Post-Fulfillment	`rating`	provide rating	acknowledgment
	`support`	request support	support info

Beckn Protocol - Discovery



Scenario 1: Privacy

Disclaimer: Not necessarily a vulnerability in the current version of Beckn

Scenario:

I make two separate searches: A taxi to my home and <SENSITIVE>

The seraches are broadcasted to all eligible sellers

I only get offers from legit-looking sellers, and simply choose two of them

Later I receive blackmail threatening to expose that I like <SENSITIVE>

Scenario 1: Privacy

Disclaimer: Not necessarily a vulnerability in the current version of Beckn

Scenario:

I make two separate searches: A taxi to my home and <SENSITIVE>

The seraches are broadcasted to all eligible sellers

I only get offers from legit-looking sellers, and simply choose two of them

Later I receive blackmail threatening to expose that I like <SENSITIVE>

Attack:

Mallory has a “shop” that “sells” (thus eligible) everything

She receives both of my searches, but does not respond to either

She correlates my identity from the taxi search, and that I like <SENSITIVE>

Scenario 1: Privacy

Disclaimer: Not necessarily a vulnerability in the current version of Beckn

Scenario:

I make two separate searches: A taxi to my home and <SENSITIVE>

The seraches are broadcasted to all eligible sellers

I only get offers from legit-looking sellers, and simply choose two of them

Later I receive blackmail threatening to expose that I like <SENSITIVE>

Attack:

Mallory has a “shop” that “sells” (thus eligible) everything

She receives both of my searches, but does not respond to either

She correlates my identity from the taxi search, and that I like <SENSITIVE>

Problem: Searches reveal BAP identity, participation is optional

Scenario 1: Privacy

Disclaimer: Not necessarily a vulnerability in the current version of Beckn

Scenario:

I make two separate searches: A taxi to my home and <SENSITIVE>

The seraches are broadcasted to all eligible sellers

I only get offers from legit-looking sellers, and simply choose two of them

Later I receive blackmail threatening to expose that I like <SENSITIVE>

Attack:

Mallory has a “shop” that “sells” (thus eligible) everything

She receives both of my searches, but does not respond to either

She correlates my identity from the taxi search, and that I like <SENSITIVE>

Problem: Searches reveal BAP identity, participation is optional
Possible Solution: Dont inform sellers of my identity until I accept their offer

Scenario 2: Availability

Disclaimer: Not necessarily a vulnerability in the current version of Beckn

Scenario:

I search for a taxi

The search is broadcasted to all eligible sellers

I get an overwhelming list of offers, most of which are irrelevant

Scenario 2: Availability

Disclaimer: Not necessarily a vulnerability in the current version of Beckn

Scenario:

I search for a taxi

The search is broadcasted to all eligible sellers

I get an overwhelming list of offers, most of which are irrelevant

Attack:

Mallory has a competing taxi company in another area

She receives my search, and responds with many offers for irrelevant taxis

My app is flooded with noise, drowning the taxi offer I actually wanted

Scenario 2: Availability

Disclaimer: Not necessarily a vulnerability in the current version of Beckn

Scenario:

I search for a taxi

The search is broadcasted to all eligible sellers

I get an overwhelming list of offers, most of which are irrelevant

Attack:

Mallory has a competing taxi company in another area

She receives my search, and responds with many offers for irrelevant taxis

My app is flooded with noise, drowning the taxi offer I actually wanted

Problem: Mallory can make offers that are effectively irrelevant

Scenario 2: Availability

Disclaimer: Not necessarily a vulnerability in the current version of Beckn

Scenario:

I search for a taxi

The search is broadcasted to all eligible sellers

I get an overwhelming list of offers, most of which are irrelevant

Attack:

Mallory has a competing taxi company in another area

She receives my search, and responds with many offers for irrelevant taxis

My app is flooded with noise, drowning the taxi offer I actually wanted

Problem: Mallory can make offers that are effectively irrelevant
Possible Solution: Determine and rule out "irrelevant" offers

Key Takeaways

Security violations in decentralized networks are subtle, e.g.

- Security violations may be seemingly unobservable
- Security violations may be semantically contingent

Key Takeaways

Security violations in decentralized networks are subtle, e.g.

- Security violations may be seemingly unobservable
- Security violations may be semantically contingent

A strict (protocol) policy is warranted

- Guides adoption, enforcement, and accountability
- Beckn has (had) conflicting documentation of their protocol

Key Takeaways

Security violations in decentralized networks are subtle, e.g.

- Security violations may be seemingly unobservable
- Security violations may be semantically contingent

A strict (protocol) policy is warranted

- Guides adoption, enforcement, and accountability
- Beckn has (had) conflicting documentation of their protocol

Our Opinion: Formal specifications should be a requirement

Key Takeaways

Security violations in decentralized networks are subtle, e.g.

- Security violations may be seemingly unobservable
- Security violations may be semantically contingent

A strict (protocol) policy is warranted

- Guides adoption, enforcement, and accountability
- Beckn has (had) conflicting documentation of their protocol

Our Opinion: Formal specifications should be a requirement
Research Question: Is this even feasible?

Key Takeaways

Security violations in decentralized networks are subtle, e.g.

- Security violations may be seemingly unobservable
- Security violations may be semantically contingent

A strict (protocol) policy is warranted

- Guides adoption, enforcement, and accountability
- Beckn has (had) conflicting documentation of their protocol

Our Opinion: Formal specifications should be a requirement
Research Question: Is this even feasible?
Our Contribution: Yes it is (to some extend)

Our Contribution: An Approach to Formality in Decentralized Networks

Demonstrated through the Beckn Protocol

Overivew of Approach

Protocol specification

- Formal model of messages
- Formal model of message requirements (assumptions)
- Formal model of interactions

Overivew of Approach

Protocol specification

- Formal model of messages
- Formal model of message requirements (assumptions)
- Formal model of interactions

Protocol guarantees

- Formally defined properties (e.g. safety, liveness, privacy)
- (In)formal proofs: Testing, deductive proof systems, mechanisation

Overivew of Approach

Protocol specification

- Formal model of messages
- Formal model of message requirements (assumptions)
- Formal model of interactions

Protocol guarantees

- Formally defined properties (e.g. safety, liveness, privacy)
- (In)formal proofs: Testing, deductive proof systems, mechanisation

Protocol conformance

- Informal: Certification, sampling, traffic analysis
- Semi-formal: Runtime monitoring
- Formal: Static analysis

A Formal Model of Messages

Conventional Type Theory:

$$\tau ::= \mathbb{Z} \mid \mathbb{R} \mid \mathsf{String} \mid \tau_1 \times \tau_2 \mid \{x_1 : \tau_1, \ldots, x_n : \tau_n\} \mid \ldots$$

A Formal Model of Messages

Conventional Type Theory:

$$\tau ::= \mathbb{Z} \mid \mathbb{R} \mid \text{String} \mid \tau_1 \times \tau_2 \mid \{x_1 : \tau_1, \ldots, x_n : \tau_n\} \mid \ldots$$

Examples of Messages (Taxi Service):

$$\begin{aligned} \text{Intent} &::= \{\mathit{fulfillment} : \text{Fulfillment}\} \\ \text{Fulfillment} &::= \{\mathit{start} : \text{Location}, \mathit{end} : \text{Location}\} \\ \text{Location} &::= \{\mathit{gps} : \mathbb{R} \times \mathbb{R}\} \end{aligned}$$

A Formal Model of Messages

Conventional Type Theory:

$$\tau ::= \mathbb{Z} \mid \mathbb{R} \mid \text{String} \mid \tau_1 \times \tau_2 \mid \{x_1 : \tau_1, \ldots, x_n : \tau_n\} \mid \ldots$$

Examples of Messages (Taxi Service):

$$\begin{aligned} \text{Intent} &::= \{\mathit{fulfillment} : \text{Fulfillment}\} \\ \text{Fulfillment} &::= \{\mathit{start} : \text{Location}, \mathit{end} : \text{Location}\} \\ \text{Location} &::= \{\mathit{gps} : \mathbb{R} \times \mathbb{R}\} \end{aligned}$$

$$\begin{aligned} \text{Catalog} &::= \{\mathit{providers} : \text{List Provider}\} \\ \text{Provider} &::= \{\mathit{id} : \text{String}, \mathit{items} : \text{List Item}\} \\ \text{Item} &::= \{\mathit{id} : \text{String}, \mathit{price} : \mathbb{Z}, \mathit{location} : \text{Location}, \mathit{eta} : \mathbb{Z}\} \end{aligned}$$

A Formal Model of Requirements

Conventional Logical Propositions:

$$P, Q ::= \mathsf{True} \mid \mathsf{False} \mid P \wedge Q \mid P \vee Q \mid \forall x \in \tau.\, P \mid \exists x \in \tau.\, P \mid \ldots$$

A Formal Model of Requirements

Conventional Logical Propositions:

$$P, Q ::= \mathsf{True} \mid \mathsf{False} \mid P \wedge Q \mid P \vee Q \mid \forall x \in \tau.\, P \mid \exists x \in \tau.\, P \mid \ldots$$

Example of Propositional Requirement (Taxi Service):

$$\begin{aligned}&\mathsf{ValidCat}\,(\mathit{int} : \mathsf{Intent})\,(\mathit{cat} : \mathsf{Catalog}) \triangleq \\ &\quad \forall \mathit{provider} \in \mathit{cat}.\mathit{providers}.\forall \mathit{item} \in \mathit{provider}.\mathit{items}. \\ &\qquad \mathtt{distance}(\mathit{int}.\mathit{fulfillment}.\mathit{start}, \mathit{item}.\mathit{location}) < \mathtt{MAX\_DIST}\end{aligned}$$

A Formal Model of Decentralized Protocols

Dependent Session Protocols:

$$
\begin{aligned}
p ::= \; & \mathbf{!}\,[i]\,(\vec{x}:\vec{\tau})\,\langle v\rangle\{P\}.\,p \mid \\
& \mathbf{?}[i]\,(\vec{x}:\vec{\tau})\,\langle v\rangle\{P\}.\,p \mid \\
& \Sigma_{a\in As}.\,p_a \mid \\
& \mathbf{end}
\end{aligned}
$$

A Formal Model of Decentralized Protocols

Dependent Session Protocols:

$$
\begin{aligned}
p ::= \; & \mathbf{!}\,[i]\,(\vec{x}:\vec{\tau})\,\langle v\rangle\{P\}.\,p \mid \\
& \mathbf{?}[i]\,(\vec{x}:\vec{\tau})\,\langle v\rangle\{P\}.\,p \mid \\
& \Sigma_{a\in As}.\,p_a \mid \\
& \mathbf{end}
\end{aligned}
$$

Example of Protocol (Any Network):

$$
\begin{aligned}
&\mathsf{BPP} \triangleq \Sigma_{G\in Gs}. \\
&\quad \mathbf{?}[G]\,(int : \mathsf{Intent})\,\langle int\rangle. \\
&\quad \mathbf{!}\,[G]\,(cat : \mathsf{Catalog})\,\langle cat\rangle\{\mathsf{ValidCat}\,int\,cat\}. \\
&\quad \mathbf{end}
\end{aligned}
$$

A Formal Model of Decentralized Protocols

Dependent Session Protocols:

$$
\begin{aligned}
p ::= \; & \mathbf{!}\,[i]\,(\vec{x}:\vec{\tau})\,\langle v\rangle\{P\}.\,p \mid \\
& \mathbf{?}[i]\,(\vec{x}:\vec{\tau})\,\langle v\rangle\{P\}.\,p \mid \\
& \Sigma_{a\in As}.\,p_a \mid \\
& \mathbf{end}
\end{aligned}
$$

Example of Protocol (Any Network):

$$
\begin{aligned}
\mathsf{BPP} \;\triangleq\; & \Sigma_{G\in Gs}. \\
& \mathbf{?}[G]\,(\mathit{int} : \mathsf{Intent})\,\langle \mathit{int}\rangle. \\
& \mathbf{!}\,[G]\,(\mathit{cat} : \mathsf{Catalog})\,\langle \mathit{cat}\rangle\{\mathsf{ValidCat}\;\mathit{int}\;\mathit{cat}\}. \\
& \mathbf{end}
\end{aligned}
$$

Treats message types and requirements abstractly!:

$$M ::= \mathsf{Intent} \mid \mathsf{Catalog} \mid \ldots \qquad\qquad R ::= \mathsf{ValidCat}\;\mathit{int}\;\mathit{cat} \mid \ldots$$

A Formal Model of the Beckn Protocol

Beckn Protocol - Discovery Phase:

$\mathsf{BAP} \triangleq \Sigma_{G \in Gs}.$
$\quad !\,[G]\,(int : \mathsf{Intent})\,\langle int \rangle.\,\mathsf{BAP}'\,G$

$\mathsf{BAP}'\,G \triangleq$
$\quad ?[G]\,(b : \mathbb{B})\,(cat : \mathsf{Catalog})\,\langle (b, cat) \rangle \{\mathsf{ValidCat}\,int\,cat\}.$
$\quad \mathbf{if}\,b\,\mathbf{then}\,\mathbf{end}\,\mathbf{else}\,\mathsf{BAP}'\,G$

$\mathsf{BPP} \triangleq \Sigma_{G \in Gs}.$
$\quad ?[G]\,(int : \mathsf{Intent})\,\langle int \rangle.$
$\quad !\,[G]\,(cat : \mathsf{Catalog})\,\langle cat \rangle \{\mathsf{ValidCat}\,int\,cat\}.$
$\quad \mathbf{end}$

$\mathsf{BG} \triangleq \ldots$
$\mathsf{BR} \triangleq \ldots$

A Formal Model of the Beckn Protocol

Beckn Protocol - Discovery Phase:

$\mathrm{BAP} \triangleq \Sigma_{G \in Gs}.$
$\quad !\,[G]\,(int : \mathrm{Intent})\,\langle int \rangle.\,\mathrm{BAP}'\,G$
$\mathrm{BAP}'\,G \triangleq$
$\quad ?[G]\,(b : \mathbb{B})\,(cat : \mathrm{Catalog})\,\langle (b, cat) \rangle \{\mathrm{ValidCat}\,int\,cat\}.$
$\quad \mathbf{if}\,b\,\mathbf{then}\,\mathbf{end}\,\mathbf{else}\,\mathrm{BAP}'\,G$

$\mathrm{BPP} \triangleq \Sigma_{G \in Gs}.$
$\quad ?[G]\,(int : \mathrm{Intent})\,\langle int \rangle.$
$\quad !\,[G]\,(cat : \mathrm{Catalog})\,\langle cat \rangle \{\mathrm{ValidCat}\,int\,cat\}.$
$\quad \mathbf{end}$

$\mathrm{BG} \triangleq \ldots$
$\mathrm{BR} \triangleq \ldots$

Beckn Protocol - Order Phase:

$\mathrm{BAP\_ord}\,(cat : \mathrm{Catalog}) \triangleq$
$\quad !\,[S]\,(sel : \mathrm{Selection})\,\langle sel \rangle \{\mathrm{ValidSel}\,sel\,cat\}.$
$\quad ?[S]\,(ord : \mathrm{Order})\,\langle ord \rangle \{\mathrm{ValidOrd}\,ord\,sel\}.$
$\quad !\,[S]\,(bil : \mathrm{Billing})\,\langle ord <| bil \rangle.$
$\quad ?[S]\,(pay : \mathrm{Payment})\,\langle ord <| bil <| pay \rangle.$
$\quad !\,[S]\,\langle ord <| bil <| pay \rangle.$
$\quad ?[S]\,(sta : \mathrm{Status})\,\langle ord <| bil <| pay <| sta \rangle.$
$\quad \mathbf{end}$

$\mathrm{BPP\_ord}\,(cat : \mathrm{Catalog}) \triangleq$
$\quad ?[B]\,(sel : \mathrm{Selection})\,\langle sel \rangle \{\mathrm{ValidSel}\,sel\,cat\}.$
$\quad !\,[B]\,(ord : \mathrm{Order})\,\langle ord \rangle \{\mathrm{ValidOrd}\,ord\,sel\}.$
$\quad ?[B]\,(bil : \mathrm{Billing})\,\langle ord <| bil \rangle.$
$\quad !\,[B]\,(pay : \mathrm{Payment})\,\langle ord <| bil <| pay \rangle.$
$\quad ?[B]\,\langle ord <| bil <| pay \rangle.$
$\quad !\,[B]\,(sta : \mathrm{Status})\,\langle ord <| bil <| pay <| sta \rangle.$
$\quad \mathbf{end}$

Protocol Properties

Protocol Consistency

- Provable property that all exchanges are consistent
- Variables, message, and requirements of receivers are met by senders
- OBS: No formal liveness, information flow, etc.

Protocol Properties

Protocol Consistency

- Provable property that all exchanges are consistent
- Variables, message, and requirements of receivers are met by senders
- OBS: No formal liveness, information flow, etc.

Examples are protocol consistency:

$$\textsc{consistent} \quad \begin{array}{l} [?[B]\,(cat : \mathsf{Catalog})\,\langle cat\rangle\{\mathsf{ValidCat}\,int\,cat\}.\,\mathbf{end};\\ \;\;![A]\,(cat : \mathsf{Catalog})\,\langle cat\rangle\{\mathsf{ValidCat}\,int\,cat\}.\,\mathbf{end}] \end{array} \quad \checkmark$$

Protocol Properties

Protocol Consistency

- Provable property that all exchanges are consistent
- Variables, message, and requirements of receivers are met by senders
- OBS: No formal liveness, information flow, etc.

Examples are protocol consistency:

CONSISTENT
$$[?[B]\,(cat : \mathsf{Catalog})\,\langle cat\rangle\{\mathsf{ValidCat}\,int\,cat\}.\,\mathbf{end};\; !\,[A]\,(cat : \mathsf{Catalog})\,\langle cat\rangle\{\mathsf{ValidCat}\,int\,cat\}.\,\mathbf{end}]$$
✓

CONSISTENT
$$[?[B]\,(cat : \mathsf{Catalog})\,\langle cat\rangle\{\mathsf{ValidCat}\,int\,cat\}.\,\mathbf{end};\; !\,[A]\,(cat : \mathsf{Catalog})\,\langle cat\rangle.\,\mathbf{end}]$$
✗

Protocol Properties

Protocol Consistency

- Provable property that all exchanges are consistent
- Variables, message, and requirements of receivers are met by senders
- OBS: No formal liveness, information flow, etc.

Examples are protocol consistency:

CONSISTENT $[?[B]\,(cat : \mathsf{Catalog})\,\langle cat\rangle\{\mathsf{ValidCat}\,int\,cat\}.\,\mathbf{end};\; !\,[A]\,(cat : \mathsf{Catalog})\,\langle cat\rangle\{\mathsf{ValidCat}\,int\,cat\}.\,\mathbf{end}]$ ✓

CONSISTENT $[?[B]\,(cat : \mathsf{Catalog})\,\langle cat\rangle\{\mathsf{ValidCat}\,int\,cat\}.\,\mathbf{end};\; !\,[A]\,(cat : \mathsf{Catalog})\,\langle cat\rangle.\,\mathbf{end}]$ ✗

CONSISTENT $[\mathsf{BAP}; \mathsf{BG}; \mathsf{BR}; \mathsf{BPP}]$ ✓

Protocol Properties

Protocol Consistency

- Provable property that all exchanges are consistent
- Variables, message, and requirements of receivers are met by senders
- OBS: No formal liveness, information flow, etc.

Examples are protocol consistency:

CONSISTENT $[?[B]\,(cat : \text{Catalog})\,\langle cat\rangle\{\text{ValidCat}\,int\,cat\}.\,\textbf{end};\; !\,[A]\,(cat : \text{Catalog})\,\langle cat\rangle\{\text{ValidCat}\,int\,cat\}.\,\textbf{end}]$ ✓

CONSISTENT $[?[B]\,(cat : \text{Catalog})\,\langle cat\rangle\{\text{ValidCat}\,int\,cat\}.\,\textbf{end};\; !\,[A]\,(cat : \text{Catalog})\,\langle cat\rangle.\,\textbf{end}]$ ✗

CONSISTENT $[\text{BAP}; \text{BG}; \text{BR}; \text{BPP}]$ ✓

CONSISTENT $[\text{BAP\_ord}\,cat; \text{BPP\_ord}\,cat]$ ✓

Protocol Properties

Protocol Consistency

- Provable property that all exchanges are consistent
- Variables, message, and requirements of receivers are met by senders
- OBS: No formal liveness, information flow, etc.

Examples are protocol consistency:

CONSISTENT $[?[B]\,(cat : \mathsf{Catalog})\,\langle cat\rangle\{\mathsf{ValidCat}\,int\,cat\}.\,\mathbf{end};\; !\,[A]\,(cat : \mathsf{Catalog})\,\langle cat\rangle\{\mathsf{ValidCat}\,int\,cat\}.\,\mathbf{end}]$ ✓

CONSISTENT $[?[B]\,(cat : \mathsf{Catalog})\,\langle cat\rangle\{\mathsf{ValidCat}\,int\,cat\}.\,\mathbf{end};\; !\,[A]\,(cat : \mathsf{Catalog})\,\langle cat\rangle.\,\mathbf{end}]$ ✗

CONSISTENT $[\mathsf{BAP}; \mathsf{BG}; \mathsf{BR}; \mathsf{BPP}]$ ✓

CONSISTENT $[\mathsf{BAP\_ord}\,cat; \mathsf{BPP\_ord}\,cat]$ ✓

Mechanised in the Rocq Prover

Protocol Conformance

Actris: Program Logic for Dependent Session Protocols

Protocol Conformance

Actris: Program Logic for Dependent Session Protocols

$$\left\{c \rightarrowtail \mathbf{!}\,[i]\,(\vec{x}:\vec{\tau})\,\langle v\rangle\{P\}.\,p * P[\vec{t}/\vec{x}]\right\}\,c[i].\mathbf{send}(v[\vec{t}/\vec{x}])\,\left\{c \rightarrowtail p[\vec{t}/\vec{x}]\right\}$$

Protocol Conformance

Actris: Program Logic for Dependent Session Protocols

$$\left\{c \rightarrowtail \mathbf{!}\,[i]\,(\vec{x}:\vec{\tau})\,\langle v\rangle\{P\}.\,p * P[\vec{t}/\vec{x}]\right\} c[i].\mathbf{send}(v[\vec{t}/\vec{x}])\,\left\{c \rightarrowtail p[\vec{t}/\vec{x}]\right\}$$

$$\left\{c \rightarrowtail \mathbf{?}[i]\,(\vec{x}:\vec{\tau})\,\langle v\rangle\{P\}.\,p\right\} c[i].\mathbf{recv}()\,\left\{w.\,\exists\vec{t}.\,w = v[\vec{t}/\vec{x}] * c \rightarrowtail p[\vec{t}/\vec{x}] * P[\vec{t}/\vec{x}]\right\}$$

Protocol Conformance

Actris: Program Logic for Dependent Session Protocols

$$\left\{c \rightarrowtail \mathbf{!}\,[i]\,(\vec{x}:\vec{\tau})\,\langle v\rangle\{P\}.\,p * P[\vec{t}/\vec{x}]\right\}\,c[i].\mathbf{send}(v[\vec{t}/\vec{x}])\,\left\{c \rightarrowtail p[\vec{t}/\vec{x}]\right\}$$

$$\left\{c \rightarrowtail \mathbf{?}[i]\,(\vec{x}:\vec{\tau})\,\langle v\rangle\{P\}.\,p\right\}\,c[i].\mathbf{recv}()\,\left\{w.\,\exists\vec{t}.\,w = v[\vec{t}/\vec{x}] * c \rightarrowtail p[\vec{t}/\vec{x}] * P[\vec{t}/\vec{x}]\right\}$$

OBS: Only applied and proven sound wrt operational semantics of simple research-centric programming languages

Conclusion

Decentralized networks are prone to security violations

- Even a lack of response can have implications
- It is not obvious what a protocol violation means

Formal protocols give a precise description

- Rigid policy for which we can prove properties and enforce conformance

Dependent session protocols is a candidate formal language

- Permits specifying dependent interactions between multiple participants
- Has infrastructure to prove protocol consistency and conformance
- Was sufficient for formalising a part of the Beckn protocol

Thank You

Questions?