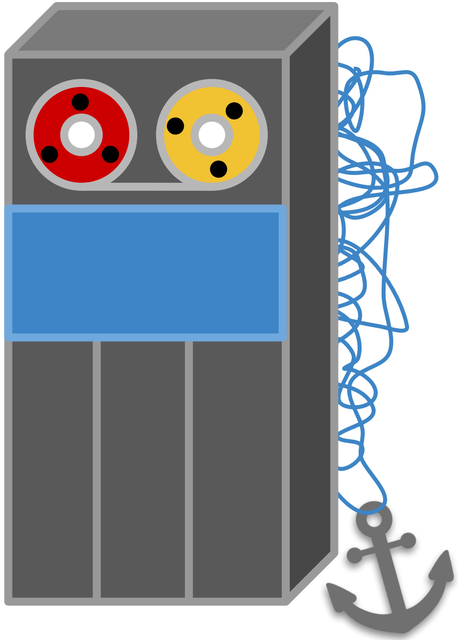
Concluding Remarks

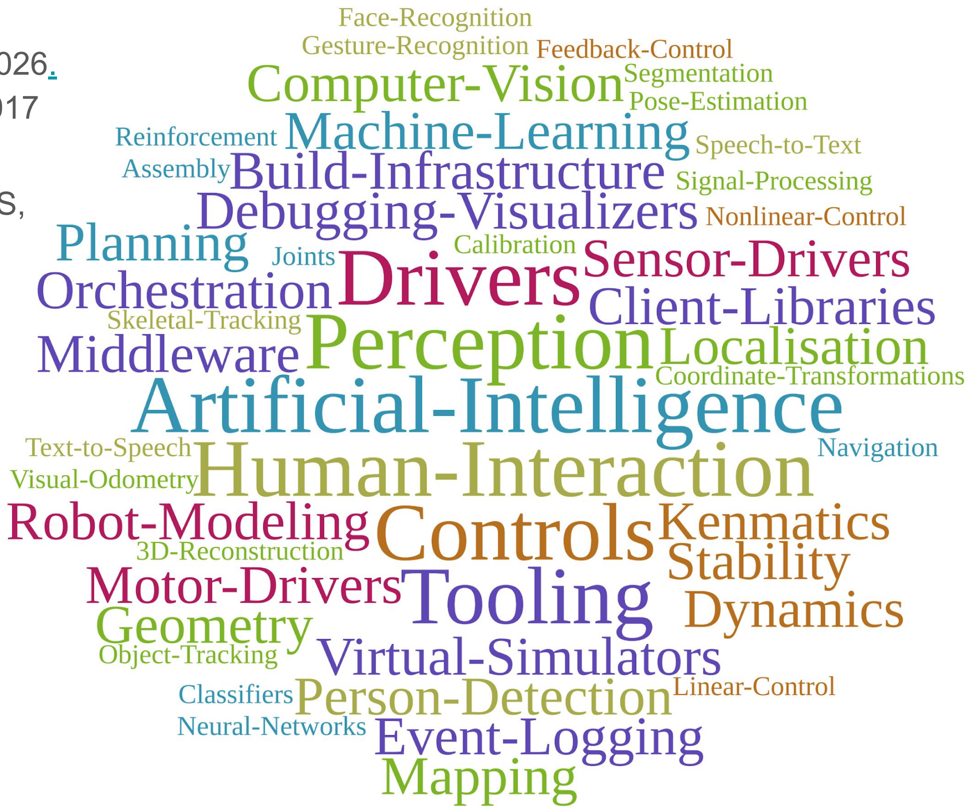


Overview

- Background
  - Summarized security issues in ROS1
- Related Work
  - What prior approaches have been proposed
- Current Work
  - Present development for SROS2
- Future Work
  - TODOs and action items for SROS2
- Conclusions
  - Closing remarks and observations

Background | Robotic Frameworks

- **ROS**
  - Market Expected to Reach US$ **402.7Mn** by 2026.
  - **+10** years development, **+13.4K** downloads 2017
- Other Examples
  - Player, YARP, Orocos, CARMEN, Orca, MOOS, Microsoft Robotics Studio, LabVIEW Robotics, MATLAB Robotics Toolbox



M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source robot operating system,” in ICRA workshop on open source software, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5

Cited **4,430** times as of 2018, up 26% from 2017

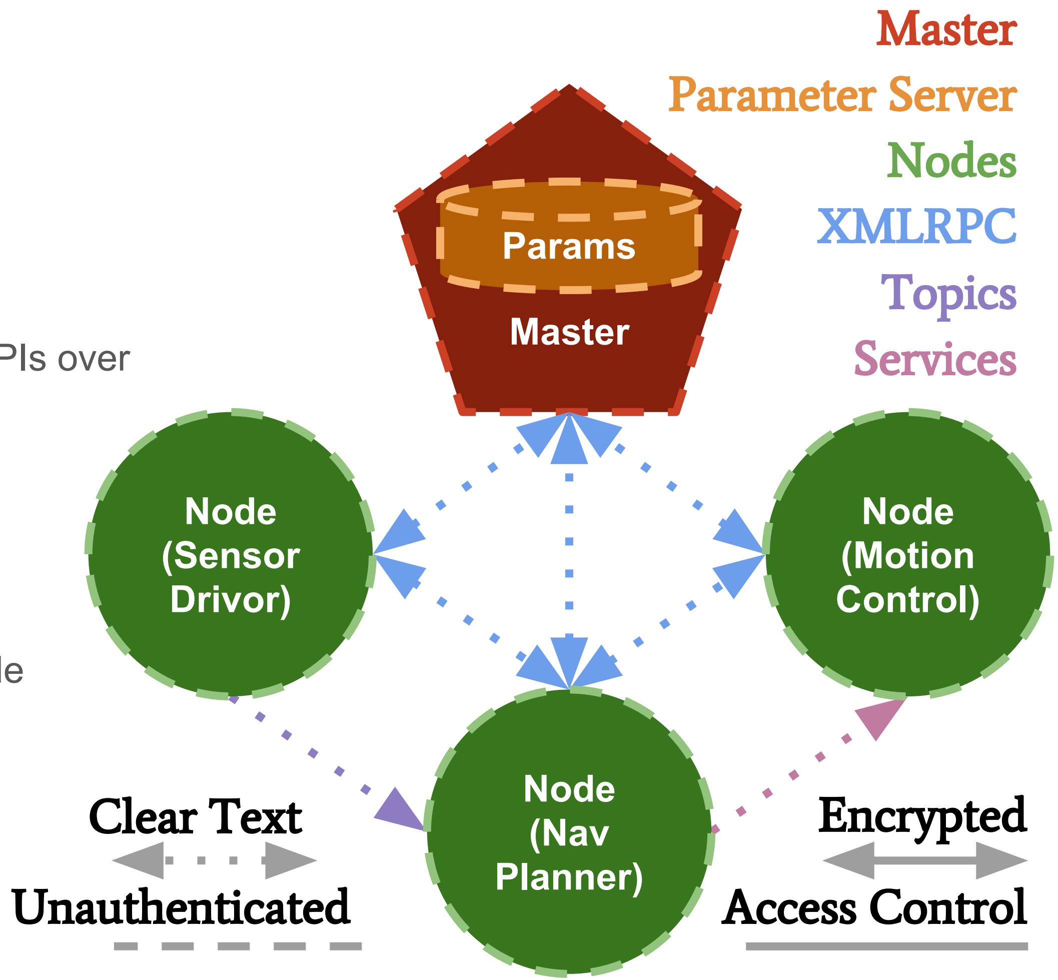
Background | ROS

Robotic Operating System (ROS)

- **Plumbing**: Middleware for process communication
- **Tooling**: Introspective debugging and visualization
- **Capabilities**: Reusable domain specific modules
- **Ecosystem:** Collaborative open source communities

Background | ROS1

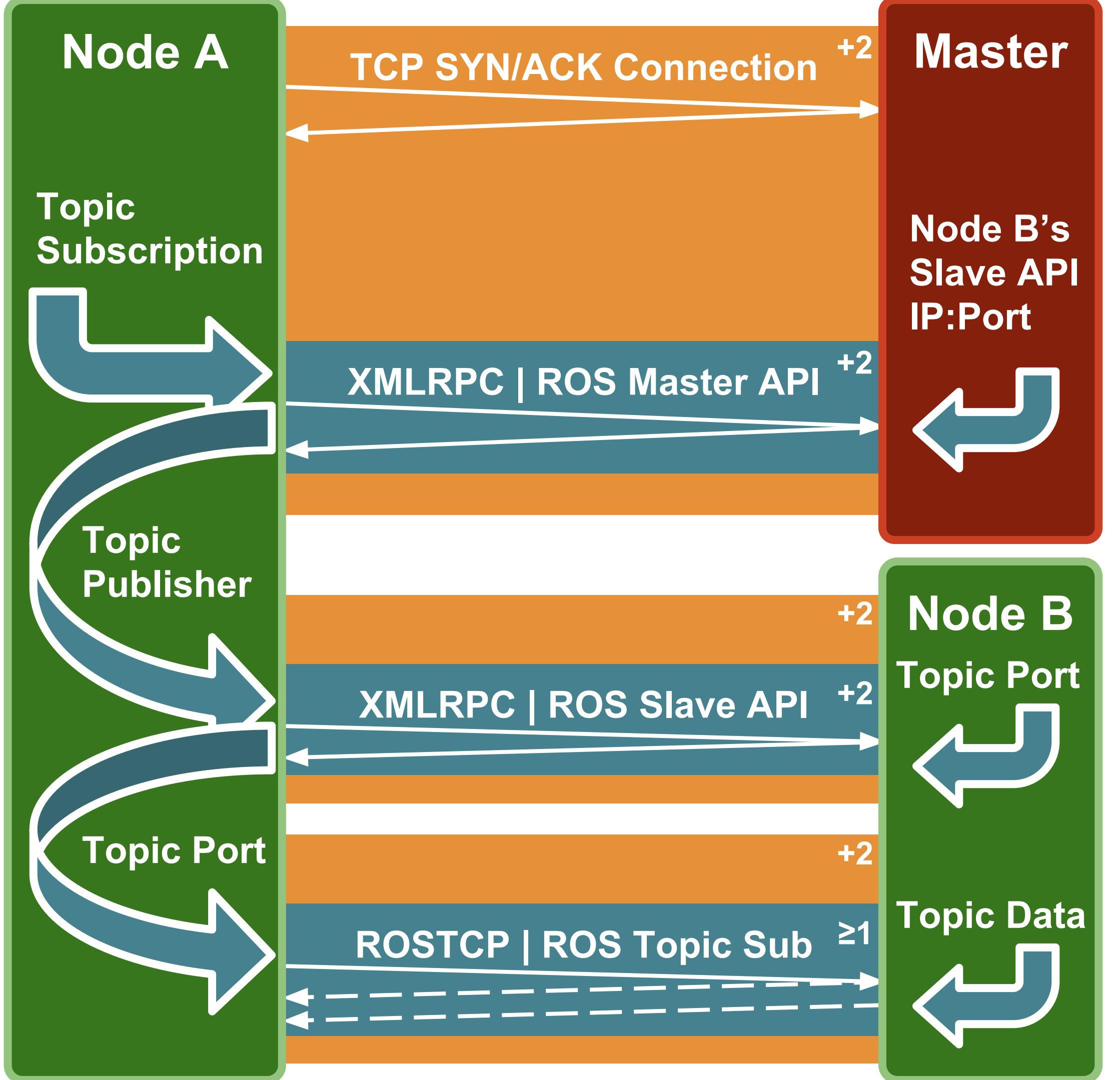
- Peer-to-peer pub/sub model formulates **anonymous** computational graph
- Processes communicate through common APIs over **clear text** transport via topics and services
- Master provides namespace resolution and **centralized** key-value parameter storage
- APIs subsystems are unregulated and provide **unauthenticated** access to any connection



Background | ROS1

Illustrated subscription in ladder diagram

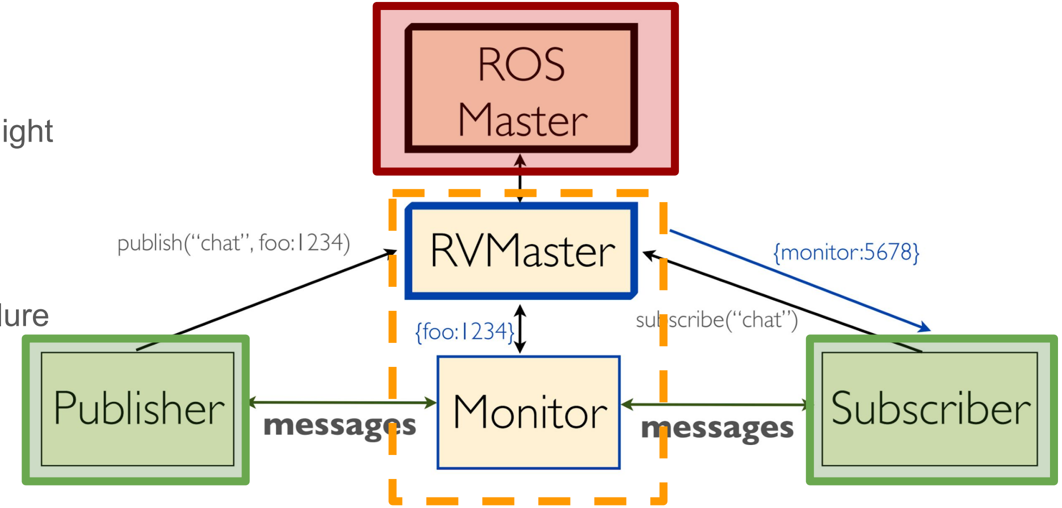
1. Node A sends topic subscription request
2. Master returns publisher list in callback
3. Node A negotiates transport method
4. Node B returns transport specifics
5. Node A connects, receives topic data



Related Work | ROS-RV

Redirecting ROS traffic through MITM mediator

- Pros
  - **Runtime Verification:** of message data in flight
  - **Compatibility**: Maintains application API
- Cons
  - **Unencrypted**: Transport level still exposed
  - **SPOF**: RVMaster adds a Single Point of Failure
  - **Scalability**: Added overhead from Monitor
  - **Access Control**: Limited to IP level
  - **Flexible**: Not suitable for dynamic networks
  - **Subsystems**: Not all APIs are protected

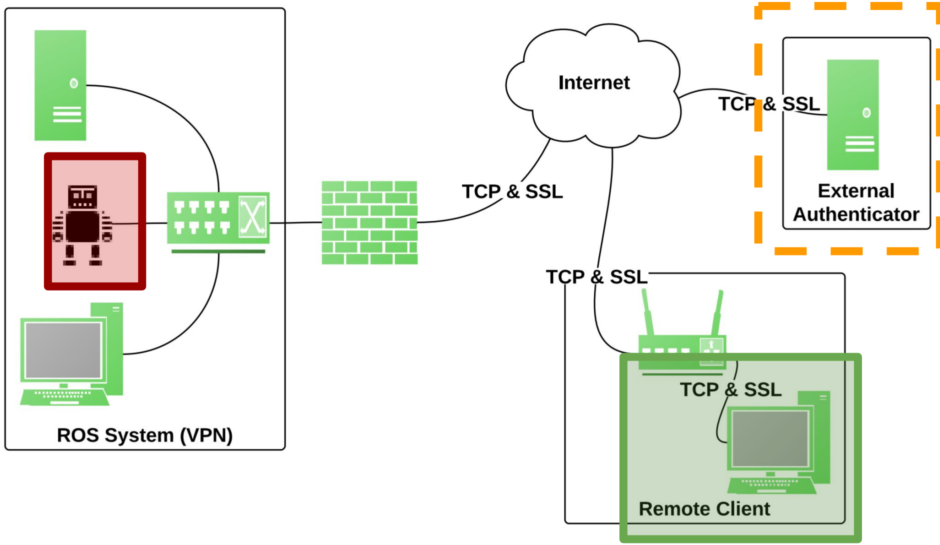


J. Huang, et. al, "Rosrv: Runtime verification for robots," in Proceedings of the 14th International Conference on Runtime Verification, ser. LNCS, vol. 8734. Springer International Publishing, September 2014, pp. 247–254.

Related Work | Rosauth

Enabling private and authentic remote connectivity

- Pros
  - **Secure Transport**: via authenticated encryption
  - **Compatibility**: Maintains application API
  - **Dynamic:** Authenticator updates Access Control
- Cons
  - **SPOF**: Authenticator adds a Single Point of Failure
  - **Access Control**: authentication but no *authorization*
  - **Subsystems**: Not all APIs are protected
  - **Limited Scope**: Non-native ROS clients only

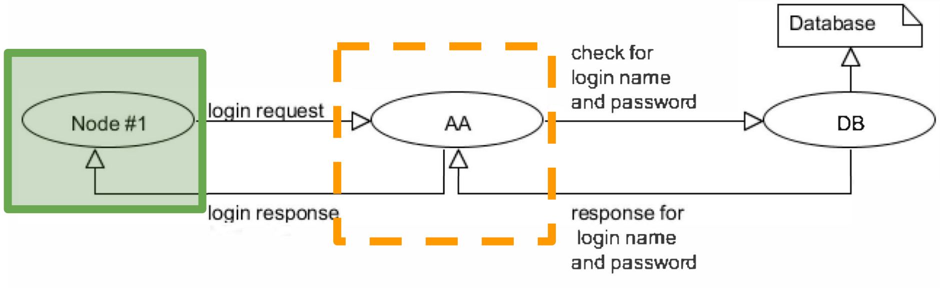


R. Toris, C. Shue, S. Chernova, “Message Authentication Codes for Secure Remote Non-Native Client Connections to ROS Enabled Robots”. In Proc. of the 2014 IEEE International Conference on Technologies for Practical Robot Applications (TePRA), Woburn, MA, USA, April 14-15, 2014.

Related Work | ROS-ALG

Application Level Gateway for key distribution

- Pros
  - **Dynamic:** DataBase updates Access Control
  - **Accounting**: Enables auditing of events
  - **Compatibility**: Maintains application API
- Cons
  - **SPOF**: AA node adds a Single Point of Failure
  - **Custom Crypto**: Rolls own transport encryption
  - **Subsystems**: Not all APIs are protected

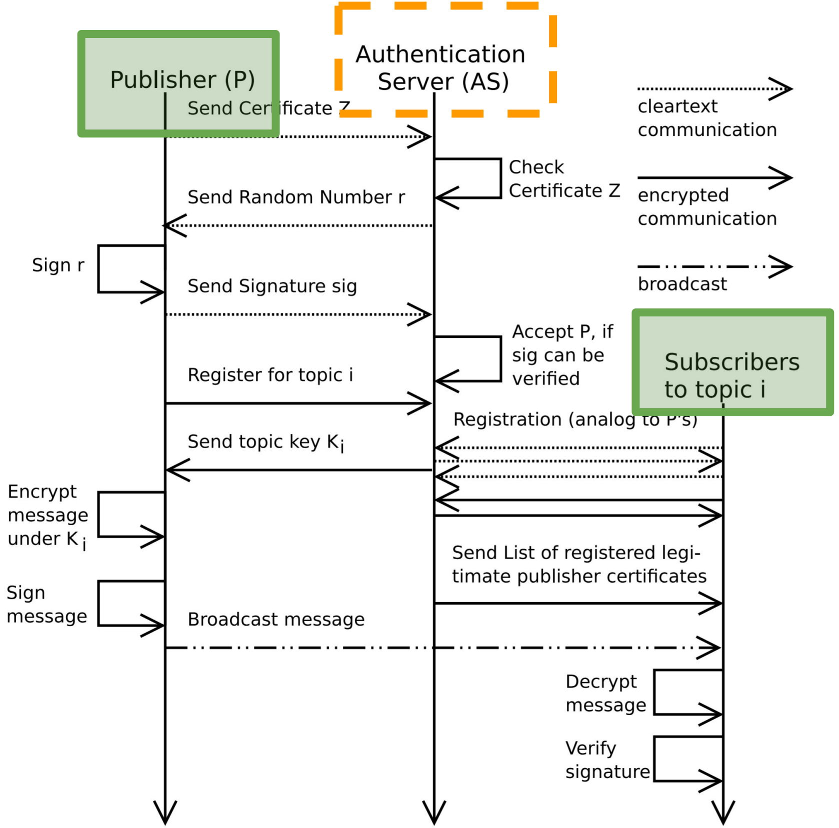


R. Dczi, et. al, “Increasing ros 1.x communication security for medical surgery robot,” in 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Oct 2016, pp. 4444–4449

Related Work | Secure-ROS-Transport

Application Level Gateway for key distribution

- Pros
  - **Secure Transport**: for topics at least
  - **ABI**: No client library modification
- Cons
  - **Compatibility**: divergent application API
  - **SPOF**: AA node adds a Single Point of Failure
  - **Custom Crypto**: Rolls own transport encryption
  - **Subsystems**: Not all APIs are protected
  - **Access Control**: authentication but no *authorization*

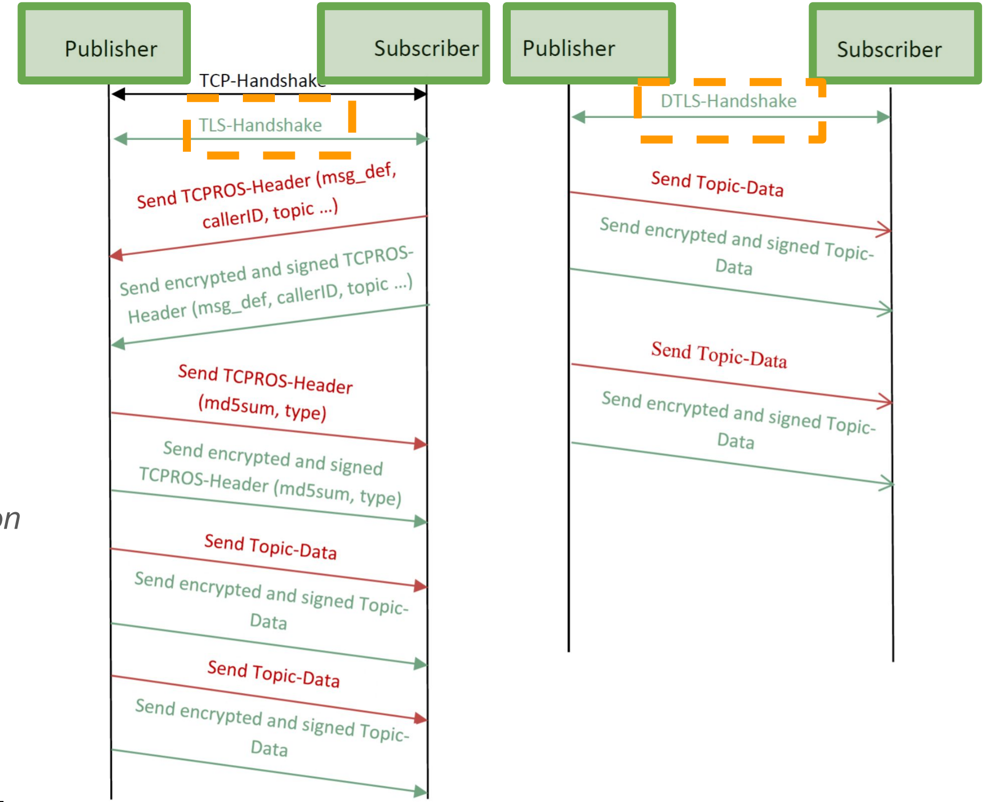


B. Dieber, S. Kacianka, S. Rass, and P. Schartner, "Application-level security for ROS-based applications," in Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on. IEEE, 2016, pp. 4477–4482.

Related Work | ROS-AES-Encryption

Decentralised authentication for transport

- Pros
  - **Secure Transport**: via authenticated encryption
  - **Standard Crypto**: Use of TLS libraries
  - **Compatibility**: Maintains application API
  - **No SPOF**: Distributed access control
  - **More QoS**: Support DTLS over UDP
- Cons
  - **Subsystems**: Not all APIs are protected
  - **Access Control**: authentication but no *authorization*
  - **Coupling**: Identity and permissions are conjoined



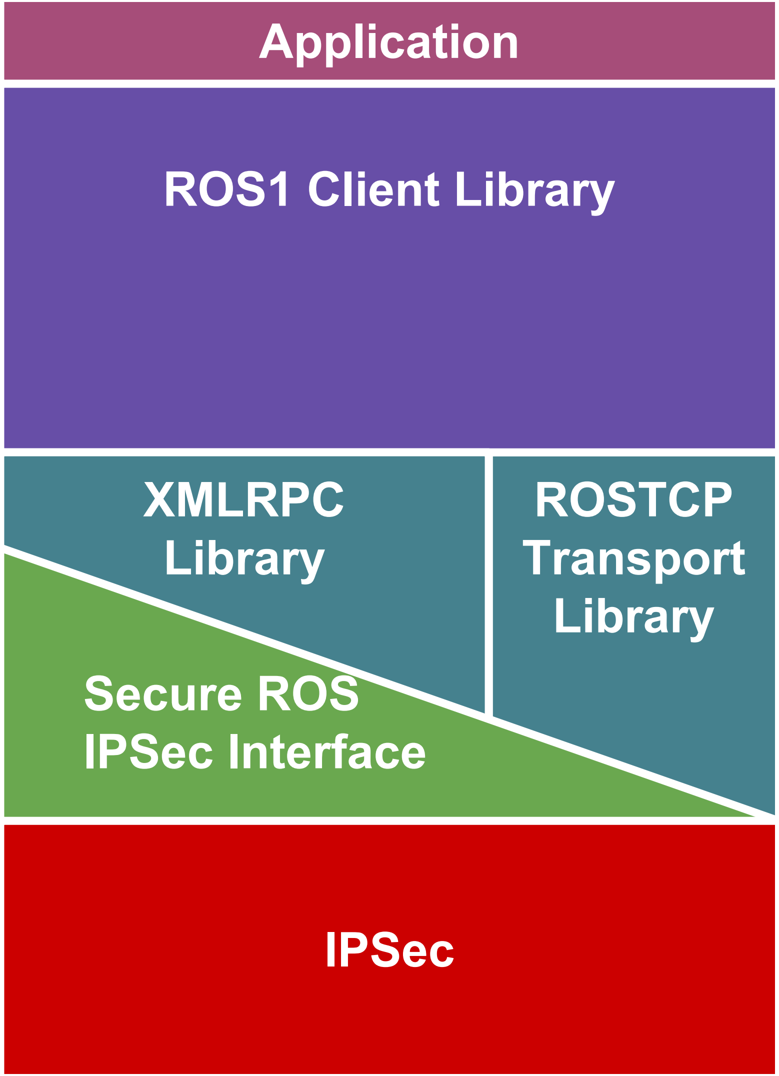
B. Breiling, B. Dieber, and P. Schartner, “Secure communication for the robot operating system,” in 2017 Annual IEEE International Systems Conference (SysCon), April 2017, pp. 1–6.

Related Work | Secure ROS

Decentralised authentication and authorization

- Pros
  - **Secure Transport**: via authenticated encryption
  - **Standard Crypto**: Use of IPSec libraries
  - **Compatibility**: Maintains application API
  - **No SPOF**: Distributed access control
  - **Coupling**: Identity/permissions are loosely conjoined
- Cons
  - **Access Control**: Limited to IP level
  - **Flexible**: Not suitable for dynamic networks
  - **Less QoS**: TCP only, so no UDP multicasting

A. Sundaresan, L. Gerard, and M. Kim, "Secure ROS: Imposing secure communication in a ROS system" 2017, ROSCon, Vancouver, Canada. [Online]. Available: https://vimeo.com/236173311

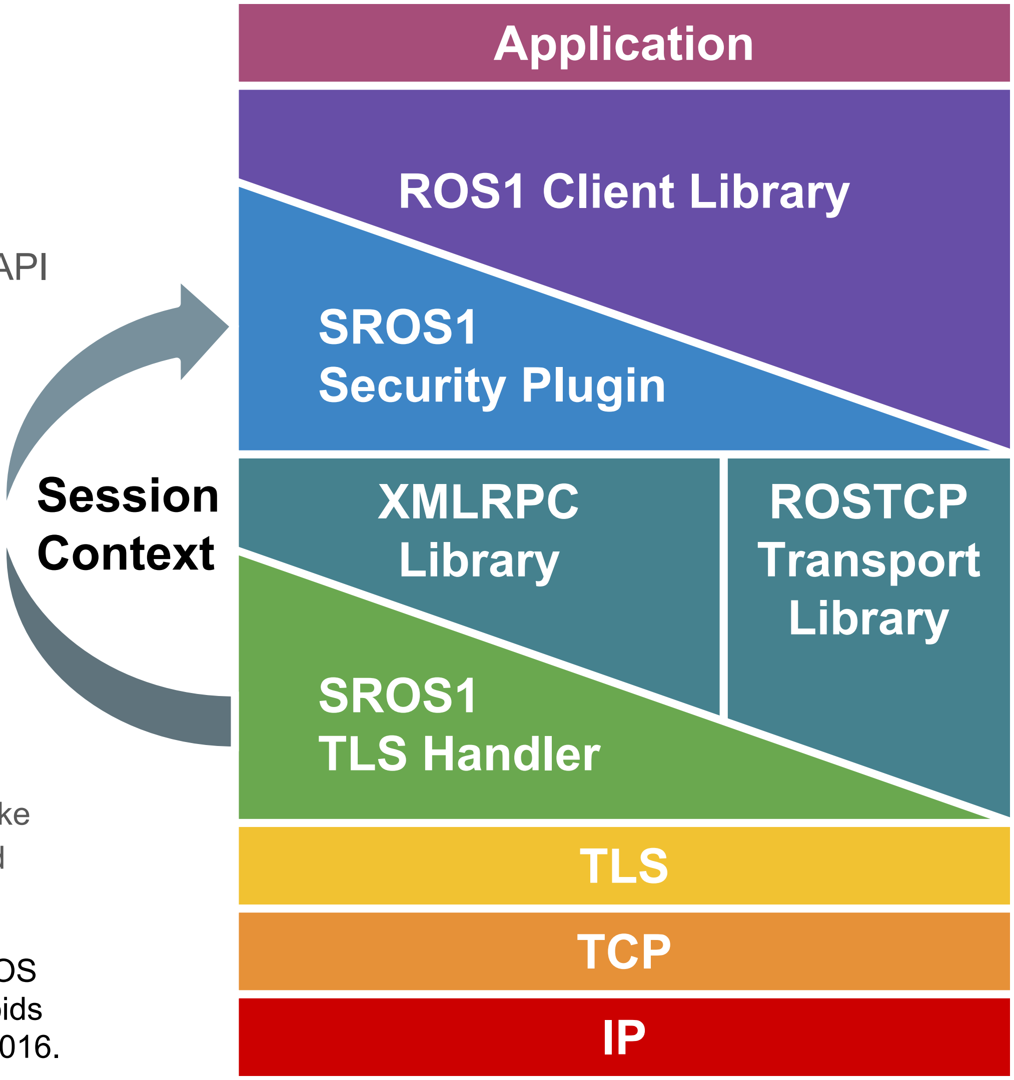


Related Work | SROS1

Decentralised authentication and authorization of full API

- Pros
  - **Secure Transport**: via authenticated encryption
  - **Standard Crypto**: Use of TLS libraries
  - **Compatibility**: Maintains application API
  - **Access Control**: Fine grained permissions
  - **Subsystems**: All APIs are guarded
  - **No SPOF**: Distributed access control
  - **Accounting**: Enables auditing of events
- Cons
  - **Context Leaking**: Access criteria embedded in identity cert publicly disclosed from TLS handshake
  - **Coupling**: Identity and permissions are conjoined
  - **Less QoS**: TCP only, so no UDP multicasting

R. White, M. Quigley, and H. Christensen, “SROS: Securing ROS over the wire, in the graph, and through the kernel,” in Humanoids Workshop: Towards Humanoid Robots OS. Cancun, Mexico, 2016.



Related Work | SROS

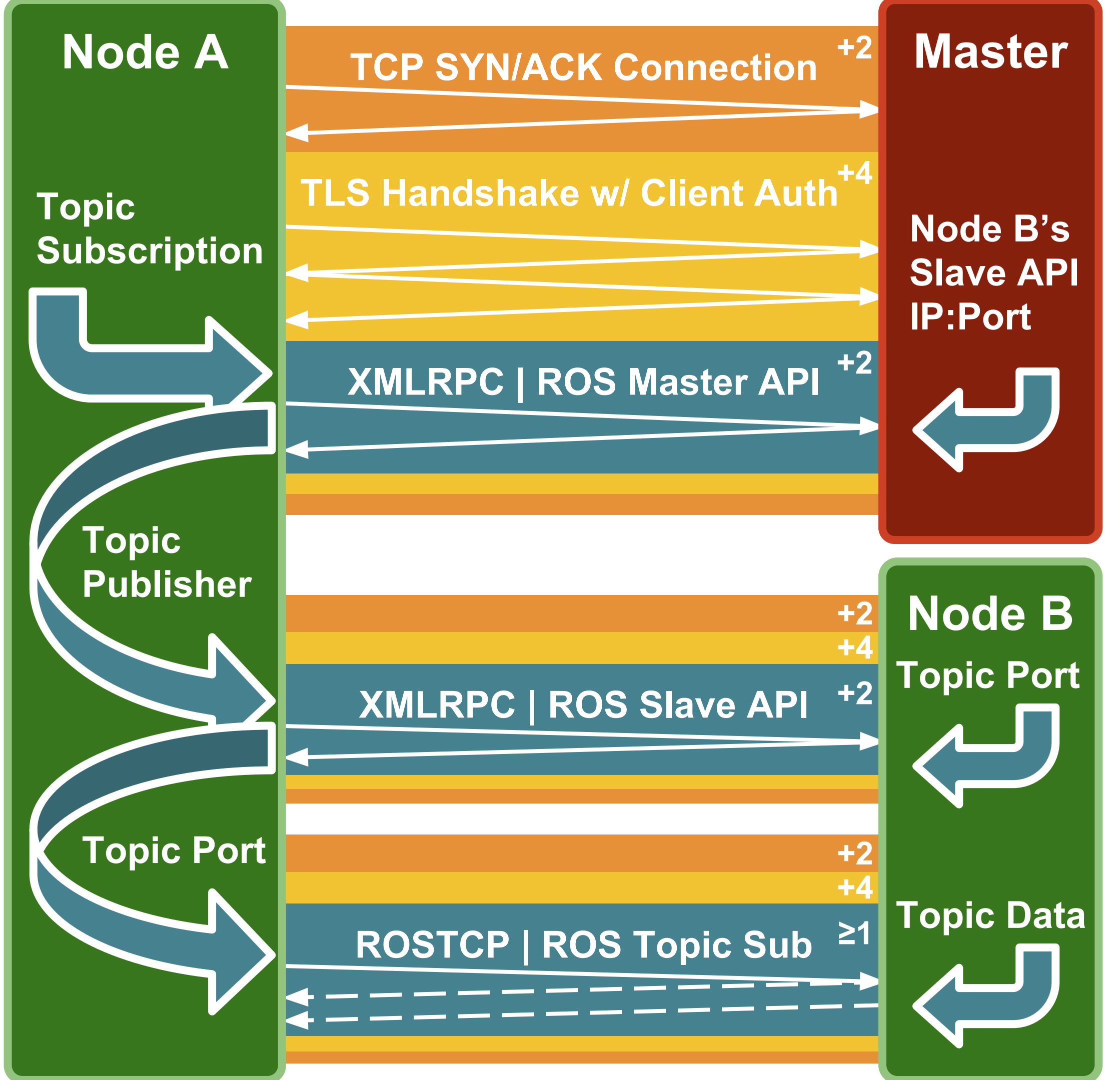
Illustrated subscription in ladder diagram

1. Node A starts TLS handshake with master, verifying API permissions before sending topic subscription request

2. Master returns sanitized publisher list in callback that Node A has permissions to

3. Node A negotiates transport method via TLS with B, gaining transport specifics.

4. Node A connects over separate TLS session and receives topic data



Criteria Approach	Encryption	Authentication	Authorization	Compatibility	Subsystems	SPOF	QoS	Scalability	Dynamic	Flexible	Accounting	Coupling
ROS-RV	None	IP	✘	API	Topic Only	✘	Rel & Best?	✘	✔	✘	✘	N/A
Rosauth	TLS	Token	✘	N/A	N/A	✘	Rel	√	✔	√	✘	N/A
ROS-ALG	Cstm+ SSH	Pass	PKI	API	Topic Only	✘	Rel & Best?	✘	✔	✔	✔	N/A
Secure-ROS-Transport	Cstm	PKI	✘	ABI	Topic Only	✘	Rel & Best?	✘	✔	✔	✘	N/A
ROS-AES-Encryption	TLS DTLS	PKI	PKI	API	Topic Only	✘	Rel & Best	✔	✔	✔	✘	Tight
Secure ROS	IPSec	PKI	IP	API	Topic ~API	✔	Rel	✔	✘	✘	✘	Loose
SROS1	TLS	PKI	PKI	API	All	✔	Rel	✔	✘	✔	✔	Tight

Related Work | Comparison

D. Portugal, M. A. Santos, S. Pereira, and M. S. Couceiro, “On the security of robotic applications using ROS,” in Artificial Intelligence Safety and Security. CRC Press, December 2017.

- SROS1 was demonstrated as most secure initiative tested previously
- Given it extensive security layer was designed to envelop the entire API surface
- However, it languished from slow performance, as it was only ever ported to rospy

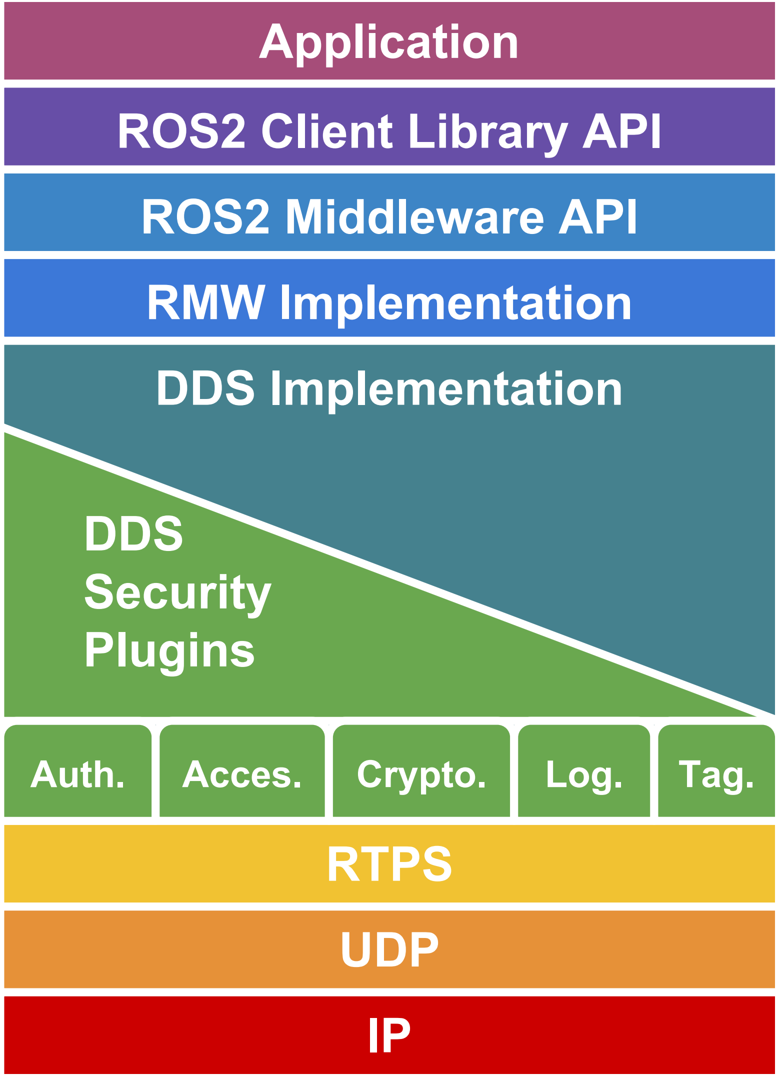
□ - Data leaked by each initiative on requests made inside a secured ROS network.

	ROS	SROS	ROS-AES-Encryption	Secure-ROS	Secure-ROS-Transport	Rosauth
rostopic list	✓	✗	✓	✗	✓	✓
rosnode list	✓	✗	✓	✓	✓	✓
rosservice list	✓	✗	✓	✓	✓	✓
rosnode kill	✓	✗	✓	✗	✓	✓
rostopic echo	✓	✗	✓✗	✗	✗	✓

Current Work | SROS2

Utilizing the DDS Security Specification

- Pros
  - **Secure Transport**: via authenticated encryption
  - **Standard Crypto**: Use of AES-GCM-GMAC
  - **Compatibility**: Maintains application API
  - **Access Control**: Fine grained permissions
  - **Subsystems**: All APIs are guarded
  - **No SPOF**: Distributed access control
  - **Accounting**: Enables auditing of events
  - **Coupling**: Identity/permissions are loosely conjoined
  - **Flexible**: Suitable for dynamic networks
  - **More QoS**: Support Sign vs Encrypt + existing QoS for DDS
  - **Plugins:** customizable for swapping or adding features
- Cons
  - **RMW Specific**: These security features are specific to RMW implementations using DDS; however security specification is standardized across DDS vendors to facilitate interoperability

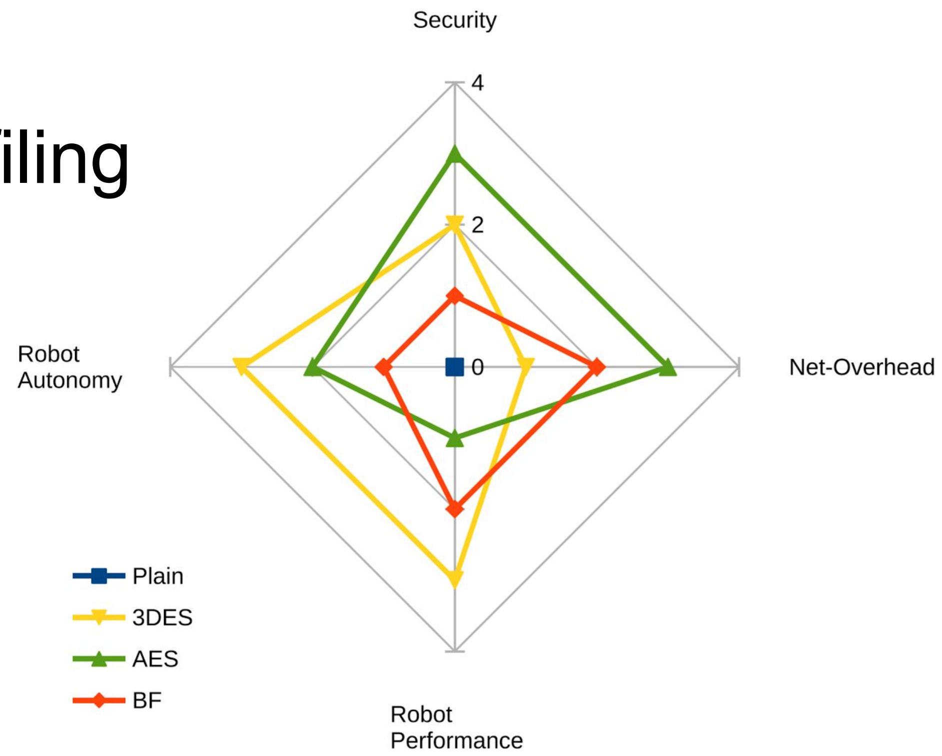


Future Work | SROS2 Profiling

- Determining optimum configurations for specific robotic deployment scenarios

- Profiling and Engineering tradeoffs
  - **Power** - energy conservation
  - **Performance** - latency
  - **Bandwidth** - network overhead
  - **Security** - cryptographic strength

- SROS2 + DDS Security
  - Embedded devices
  - Real Time systems
  - Wireless links
  - Resilient orchestration



F. J. Rodrguez-Lera, V. Matelln-Olivera, J. Balsa-Comern, . M. Guerrero-Higueras, and C. Fernndez-Llamas, “Message encryption in robot operating system: Collateral effects of hardening mobile robots,” Frontiers in ICT, vol. 5, p. 2, 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/fict.2018.00002

Future Work | SROS2 Action Items

- Finer ROS2 subsystem security
  - Instance level parameter access control
  - Preparing for upcoming ROS2 Actions
  - Hierarchical comms: robot -> fleet -> swarm
- Development and Debug Tooling
  - Assistive permission policy generation
  - Static and runtime security profiling
  - Descriptive connectivity manifests
- Management and Orchestration
  - Procedural provisioning security artifacts
  - Expressive security policy definitions
  - Generation, deployment, revocation of PKI
- Auditing and Logging
  - Distributed logging over networks
  - Recording Security Events levels
  - Cryptographically immutable records
- Trusted Execution Environments
  - Secure DDS support using TEE
  - Sealing/storage of private PKI keys
  - Protecting runtime session keys
  - E.g. Intel SGX, ARM TrustZone
- Security Testing
  - Adding additional automated CI tests
  - Static verification and code analysis
- Upstream DDS Security Issues
  - Leaking of permissions: DDSSEC12-13
  - Data-tag expressions: DDSSEC12-19
  - Instance-Level AC: DDSSEC12-12
  - Lightweight permission serialization
  - Instance vs monolithic permission exchange

Conclusions

Robots, as cyber physical systems, present a host of new security issues. However, the robotic middleware itself needn't always be a primary issue.

However, this residual security issue in robotics originate and persist from present deficiencies in:

- Tooling
  - Making security accessible
- Usability
  - Encourage user adoption
- Standardization
  - Facilitate interoperability