ROSのセキュリティ設定におけるリアルタイム性に対する考察

亘理 克好[1,a)]　大久保 隆夫[1,b)]

概要：ロボット産業は 2035 年には 9.7 兆円規模へ成長すると予想されており，産業用ロボットだけでなく，サービス分野のような新しい分野の成長が期待されている．ロボットの増加は，新たなサイバー攻撃のターゲットとされる可能性があり，特に人の近くで動作するようなサービスロボットには深刻な問題になりかねない．ロボットは様々なハードウェアによる様々な機能を持つため，開発にはミドルウェアがよく使用される．本稿では，ミドルウェアの中でも広く使用されている“ロボットオペレーティングシステム（ROS）”に着目する．ROS2 はセキュリティに対応しており，通信の暗号化などがサポートされているが，暗号化などの処理はリアルタイム性を損なう場合もある．ROS2 暗号化におけるリアルタイム性を考慮するための指標について考察し，暗号機能の設定時にリアルタイム性を考慮すべきであることを示す．

キーワード：ロボットオペレーティングシステム，ROS，ROS2，ミドルウェア，セキュリティ

Consideration of real-time performance of ROS security settings

Katsuyoshi Watari[1,a)]　Takao Okubo[1,b)]

Abstract:
The robot industry is expected to grow to a scale of 9.7 trillion yen by 2035. The increase in robots may be the target of new cyber attacks, which can be a serious problem for robots that operate near humans. Middleware is often used for robot development. In this paper, we focus on the ”robot operating system (ROS)”, which is widely used in middleware. Although ROS2 supports security functions, processing such as encryption may impair real-time performance. Consider the index for considering the real-time property in ROS2 encryption, and show that the real-time property should be considered when setting the encryption function.

Keywords: Robot Operating System, ROS, ROS2, Middleware, Security

1. はじめに

近年，様々な分野でロボットの活躍が期待されている．

それに伴い，このようなロボットの増加は新たなサイバー攻撃のターゲットとされる可能性があり，特に人の近くで動作するようなサービスロボットへの攻撃は人体や周りの設備などへの深刻な問題になりかねない．

ロボットの開発に際しては，様々な機能を持つロボットに対し，より容易に開発を可能にするロボット用のミドルウェアを使うことが主流になってきている．

また，製品やサービスに対するオープンソースソフトウェアの活用は年々増加しており，個人や大学だけでなく企業にとってもオープンソースソフトウェアは必要不可欠なものとなっている [1]．

本研究では，ロボット用ミドルウェアの中でも最も広く利用されているオープンソースソフトウェア「ロボットオペレーティングシステム（ROS）」に着目する．特に，次期バージョンの ROS2 はセキュリティに対応しており，通信の暗号化などがサポートされているが，暗号化などの処

[1] 情報セキュリティ大学院大学
Institute of Information Security
[a)] dgs198102@iisec.ac.jp
[b)] okubo@iisec.ac.jp

理はリアルタイム性を損なう場合もある．ROS2 暗号化におけるリアルタイム性を考慮するための指標について考察し，暗号機能の設定時にリアルタイム性を考慮すべきであることを示す．

2. ROS2 概要

ROS2 とこれまでの ROS(以降 ROS1 と呼ぶ) の違いとしては表 1 のようになっている．ROS1 との互換性は持っておらず，ROS1 と ROS2 の間の通信をサポートするブリッジが用意されている．また，通信方式が大きく変わっており，DDS (Data Distribution Service) を使用する．

表 1 ROS1 と ROS2 の比較
（引用：ROS ロボットプログラミングバイブル）

	ROS1	ROS2
プラットフォーム	Ubuntu, macOS	Ubuntu, macOS, Windows
通信	XMLRPC+ROSTCP	DDS
言語	C++03, Python2	C++14, Python3(3.5+)
ビルドシステム	Rosbuild → catkin	ament
メッセージサービス	*.msg, *.srv	new *.msg, *.srv, *.msg.idl, *.srv.idl
roslaunch	XML	Python
複数ノード対応	1 プロセスに 1 ノード	1 プロセスに複数ノード
グラフ表現	Remapping は起動時のみ	Remapping は実行時も可能
ネームサービス	マスタ	マスタなし (DDS ミドルウェア)

DDS は OMG で規格化されている通信ミドルウェアであり，複数の実装があるが，特に指定しない場合はデフォルトでは FastRTPS が使われる．

他にはプラットフォームに Windows が含まれていること，DDS の使用に伴い ROS マスタノードが使われなくなっていることが大きな違いである．

また，デフォルトでセキュリティをサポートし，環境変数の設定によりセキュリティ機能の ON/OFF が可能である．サポートされる主なセキュリティ機能は，通信の暗号化およびノード間のアクセス制御である．

ROS2 は商用利用されることを想定し再設計され品質向上が図られている．自動車の自動運転のプラットフォームとして ROS2 を利用し，ISO26262(機能安全) の取得を行おうとしているものもあり [2]，ROS2 を採用する製品も増えてくると考えられる．

3. ROS2 のパフォーマンス

ROS2 のパフォーマンスに関する研究としては，組み込み機器の観点からパフォーマンスを測定したものがある [3]．これは用途からリアルタイム性について，DDS の複数の実装の比較を行ったものである．ローカル PC 内に閉じたノードだけでなくリモート PC 間や ROS1 と ROS2 の混在の場合についても考慮したテスト条件になっている．DDS の実装により性能が異なることが明らかになり，それらを踏まえて実際にどの DDS 実装を使用するかを検討する必要があると結論づけている．

また，ROS2 で使用している通信ミドルウェアである DDS 単体に関しては，Pardo らは暗号化に対するオーバヘッドを計測しており [4]，1 %から 40 %のオーバヘッドがあるとの結果を示している．

DDS に限らずパブリッシュ/サブスクライブシステム全体のセキュリティに関しては，Christian らは 1998 年から 2014 年までの学術分野における文献の調査と分析を行なっている [5]．学術文献でのセキュリティ対策と製品レベルのセキュリティ対策にはまだギャップがあるとし，暗号化などのセキュリティ対策はオーバーヘッドになることから，どのようなセキュリティ対策が最良であるかを決定するための適切なセキュリティ評価手段を持つことが大切であるとしている．

最近では，Belguith らは悪意のあるサブスクライバを無効化する手法を提案している [6]．パブリッシュ/サブスクライブシステムの機能に影響を与えず，新しい鍵の再生成や再配布を行うことなく実現できる手法である．

プライバシーの観点から，Shikfa らはコンテンツベースのパブリッシュ/サブスクライブネットワークにおけるプライバシー問題を取り上げ，暗号化ルーティングテーブルを使うことで，エンドユーザ間で内容を明かさないまま通信のルーティングを行う方法を提案しているものもある [7]．

3.1 ROS2 暗号通信テスト

これまでの研究では，ROS2 の平文での通信遅延を計測したものや，DDS 単体での暗号通信測定を行ったものはあるが，ROS2 の暗号通信についてのものはなかった．通信の盗聴を想定した場合，暗号通信はその対策として使用されるが，暗号化による通信速度低下はロボットの動作に影響を与える可能性がある．そこで，筆者らはこれまで ROS2 で暗号通信を行った場合の暗号通信について通信時間の計測を行っている [8]．

また，ROS2 における OS による差異やローカル PC 間，リモート PC 間に対する暗号通信の性能差異を評価している [9]．結果の一部を**図 1** に示す．

ノード間の通信時間に着目するため，ロボットシステムとしてのノード構成ではなく，データをパブリッシュ/サブスクライブするのみのシンプルな機能を持ったノードを用意し，他のノードは実行しない状態でテストを行った．また，OS による違いの確認として，Linux，MacOS，Windows を用意しそれぞれの OS 上で ROS2 ノードを実

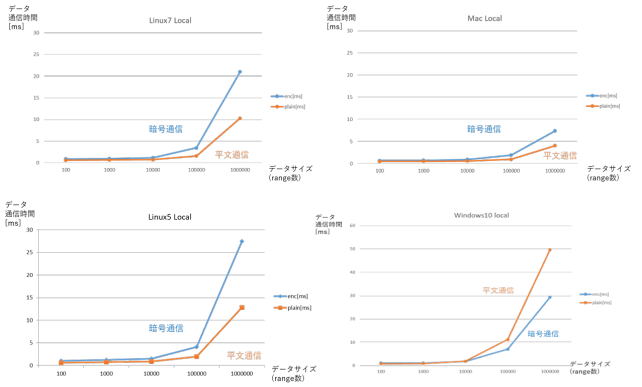

図 1 ROS2 におけるローカル PC 上での平文と暗号での通信時間結果 (左上:Linux7, 右上:MacOS, 左下:Linux5, 右下:Windows)

Fig. 1 Results of plaintext and encrypted communication time on the same PC in ROS2 (UL:Linux7, UR:MacOS, LL:Linux5,LR:Windows)

行しテストを実施した．暗号化については ROS2 のデフォルトである TLS を使用し，暗号に使うキーはパブリッシャとサブスクライバのノード毎に作成し PC 内のディレクトリにファイルとして保存した．

結果から，データサイズが増えるような場合は特に通信時間をよく検討した上で暗号化通信を使用することが重要であると言える．また，ノードの増加につれてノード間の通信も増えていくと考えられることから，セキュリティのためにすべてを暗号化するのではなく，それぞれのロボットシステムに合わせて性能とセキュリティリスクから適切に選択することが大切といえる．

4. ROS2 想定システム

ROS2 を使ったシステムとして我々は**図 2** のような Turtlebot3 を用いた自律移動システムを想定している．このロボットは本体に 2 次元 Lidar (Light Detection and Ranging) を持ち，SLAM を用いて周辺の地図を作成する．その後，ユーザから指示された目的地まで自律移動する．

このシステムでは，トピックで 32 種類，サービスで 61 種類の ROS2 データが様々な周期でやり取りされる．前述したようにこれら全てを暗号化して送信すると，ロボットの動作に影響がある可能性があるため，必要なデータについて暗号化を行うのが望ましい．

システムに対してセキュリティ対策を行う場合，通常はセキュリティアセスメントを実施しリスクに応じて対策を行う．セキュリティアセスメントには複数の手法があるが，手法によっては複雑で時間のかかるものもあるため，ROS2 の暗号機能にフォーカスし，より容易な手法で必要データを選定できるようにしたい．例えば，**図 3** のような構成パターンを想定し，ネットワーク構成と通信データの重要度などのパラメータのみで選定ができるような手法を検討する必要があると考える．

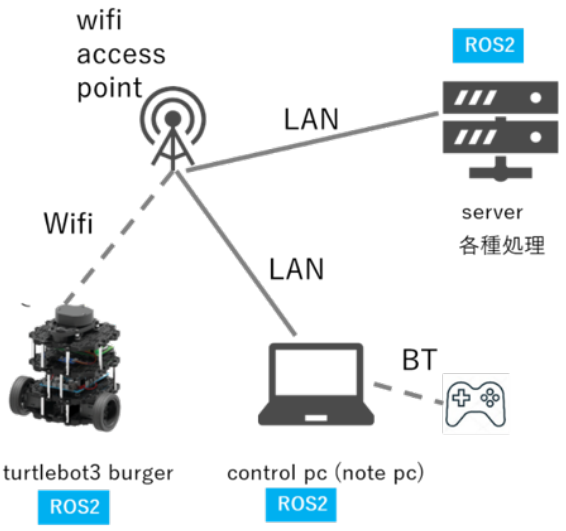

図 2 Lidar を用いて周囲の状況を把握し移動する ROS2 の自律移動システム

Fig. 2 ROS2's autonomous mobility system that uses Lidar to understand around the surroundings

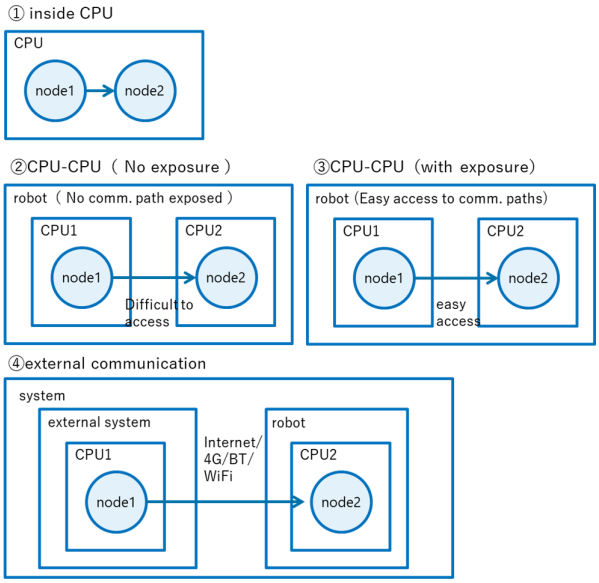

図 3 ロボットシステムにおけるネットワーク接続パターン

Fig. 3 Network connect pattern in robot system

5. サイバーセキュリティリスクアセスメント

5.1 脅威分析手法

セキュリティリスクアセスメントは，システム構築の上位である要求段階で行われる．リスクアセスメントの流れを**図 4** に示す．一般的にアセスメントは，リスクの特定とリスクの評価を順に行いリスクを決定する．リスクの特定では，被害分析と攻撃分析を行い，被害シナリオと攻撃シナリオを決定する．その結果から影響度と攻撃可能性を鑑みリスクを評価する．評価したリスクに対し，リスク対応としてリスクに合わせた方針を決定しセキュリティ要求を導出する．このようなリスクの特定と評価については様々

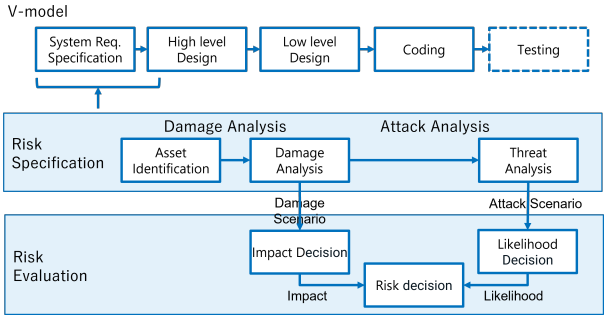

図 4 セキュリティリスクアセスメントの流れ

Fig. 4 Security risk assessment flow

表 2 主なリスクアセスメント手法

Table 2 Main risk assessment methods

Risk Specification method	Risk Evaluation method
Attack Tree	CVSS
STRIDE	CC/CEM
HEAVENS	EVITA
TM-STRIDE	HEAVENS
THROP (J3061)	CRSS
HAZOP based	RSMA
5W (TP15002)	TM-STRIDE
RWX	
TRVA	
OCTAVE	

な手法が利用されており，主な手法の一覧を**表 2** に示す．このような手法に加え，遠隔手術を行えるような遠隔ロボットにおいて，脅威を網羅的に特定するだけでなく，実際に攻撃が行われている間に手術ロボット操作者がどのような行動をとるかを明らかにするような研究も行われている [10].

5.2 Attack Tree

Attack Tree はシステムに起こりうる脅威について実際にその可能性を明確化するものである．それぞれの脅威をトップ事象として，具体的な手段を FTA のように分解し，脅威が実現されるかどうかを検討する手法である．Attack tree の例を**図 5** に示す．この例では金庫を開けるという脅威をトップ事象として，それを達成するための具体的な手法として，ピッキングや鍵の番号を知る，などと展開していく．末端事象としては，脅迫や賄賂などの事象が抽出されている．

5.3 5W method

5W 法は主に自動車分野におけるセキュリティ分析で用いられることが多い手法である．システマティックに分析を進めることができるためスキルによらず網羅的に分析をできる可能性があるが，組み合わせ爆発を起こしやすく数十万件の脅威が抽出されることもあり，その場合分析に多

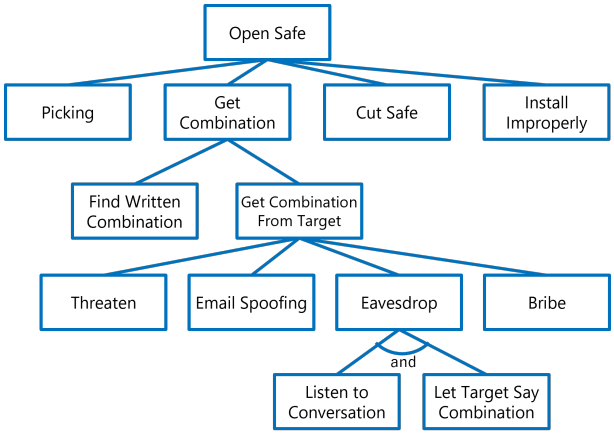

図 5 金庫が開けられる場合の Attack tree の例 (参照：Practical IoT security lecture materials)

Fig. 5 Example of attack tree (ref. Practical IoT security lecture materials)

くの時間が必要となる．具体的には，システムのアクター，資産，エントリポイントを抽出し 5W(Where, Who, When, Why, What) の組み合わせにより分析を行っていく手法である．

5.4 CVSS

CVSS(Common Vulnerability Scoring System) は IT 分野においては代表的なセキュリティリスク評価手法である．FIRST(Forum of Incident Response and Security Teams) が管理しており，CVE や JVN などの評価にも用いられている．実績のある手法として何度か改良がされており，現在のバージョンは 3.1 である．また．自動車やロボティクスへの適用については課題があるとして，様々な改良の提案が行われている．CVSS は．Base(基本評価基準)，Temporal(現状評価基準)，Environment(環境評価基準) の 3 つのメトリックグループで構成されている．基本評価基準は脆弱性そのものの深刻度を評価する基準である．システムに求められる 3 つのセキュリティ特性 (Confidentiality, Integrity Availability) に対する影響を，ネットワーク先から攻撃できるかどうかなどの基準で評価する．現状評価基準は，脆弱性が公開された後の状況により現時点での深刻度を評価する基準である．攻撃コードの有無や修正バージョンの存在により変化する基準である．環境評価基準は，エンドユーザ側の環境も含めた最終的な深刻度を評価する基準である．組織での製品の使用状況や 2 次被害などを考慮したものであり，脆弱性への対応を決定するために使われる基準となっている．例えば，基本評価基準では**表 3** にあるような基準に基づき，決められた計算式でスコアを計算する．例として Attack vector に対する区分とスコア値を**表 4** に示す．Base Metrics，Temporal Metrics，Environmental Metrics を順に計算することで最終的な深刻度を 0(Low) – 10(High) の数値で表す．スコアに対する

表 3 CVSS の基本評価基準
Table 3 CVSS base Metrics

Likelihood	Attack Vector	AV
	Attack Complexity	AC
	Privileges Required	PR
	User Interaction	UI
Impact	Confidentiality Impact	C
	Integrity Impact	I
	Availability Impact	A
	Scope	S

表 4 指標の具体的数値（AV の場合）
Table 4 Metric numerical value (AV)

Metric	Metric Value	Numerical Value
Attack vector(AV)	Network (N)	0.85
	Adjacent Network(A)	0.62
	Local (L)	0.55
	Physical (P)	0.2

表 5 CVSS スコアにおけるレベル分け
Table 5 CVSS Score rating

Rating	CVSS Score
None	0
Low	0.1〜3.9
Medium	4.0〜6.9
High	7.0〜8.9
Critical	9.0〜10.0

レベル分けについては**表 5** のようになっている．

5.5 RVSS

前述のように CVSS は様々な改良案が提案されており，そのうちの一つとして IoT やロボティクスへの適用を想定した RVSS(Robot Vul-nerability Scoring System) が提案されている [11]．RVSS では CVSS を IoT，ロボティクスに適用するにあたり，ロボットの脆弱性の重要度を正確に把握できることはできないとし，下記の点を考慮した検討を行っている．

A　ロボットの安全性の側面
B　特定の脆弱性の下流への影響
C　ライブラリ，3rd party のスコア評価
D　脆弱性の開示または Web 公開されてからの時間経過

これらの検討から，主に以下の改良を提案している．

(1) Attack Vector に対して，ロボット内の異なるネットワークタイプからの攻撃を考慮したメトリクスの追加
(2) 脆弱性が最初に報告されてからの期間を Year(Y) として metrics を追加
(3) Impact に CIA では表現できない Safety metrics を含める

			データサイズ		
			小	ー	大
		閾値	<40KB	≒40KB	>40KB
通信周期	速	>10Hz	±0	-1	-2
	|	≒10Hz	+1	±0	-1
	遅	<10Hz	+2	+1	±0

図 6 データサイズと通信周期による暗号化適用に対するリスク値への重みづけ
Fig. 6 Weights for risk values for encryption application by data size and communication cycle

6. ROS2 の暗号化適用に関する提案手法

通常，通信を暗号化するかしないか等のシステムのセキュリティ対策は前節のような様々に存在する脅威分析手法を用いてリスクを検討した上で決定される．

ロボット等の実体が動作するようなシステムの場合，セキュリティ対策をそのまま実施すると，暗号化による通信遅れなどにより，システムとしての動作が成り立たない場合も考えられる．

そこで，セキュリティリスクに対し，暗号通信の実施不実施を決めるために，通信データの周期やサイズといった情報を基に，リアルタイム性を考慮した重みづけを行う手法を提案する．

重みづけは，データの送信周期とデータサイズのマトリクスとして**図 6** のように算出し，この値をセキュリティリスクに対して加えることで，リアルタイム性を考慮したリスク値を算出し，暗号通信するかどうかを決定する．

図の閾値は，我々の行った ROS2 暗号通信結果を考慮して算出している．我々の結果では，通信データサイズが 40KB 程度までは暗号により 1.5 倍の通信時間となっており，それを超えると通信時間が増加していく．

また，データサイズが 40KB 程度では通信時間に 5ms 程度の時間が必要であり，各ノードでは通信以外の処理を行いながら実行されることを考慮し，10Hz 程度を閾値としている．

図 2 に示した我々の想定しているシステムにおいて，前述したように実際にノード間で通信されているトピックは 32 種類あり，通信周期もトピック毎に設定されている．抜粋したトピック一覧を**表 6** に示す．

このようなトピックにおいて，情報漏えい，偽メッセージ，盗聴などの脅威に対して，通信データの暗号化が行われる．

例えば具体例として，表 6 のトピックでは，/cmd_vel トピックはロボットに対する移動指令を行うときによく使われるトピックであり，偽メッセージによる攻撃はロボットの意図しない動作につながることから影響も大きいため，リスク値が高く算定されることが考えられる．また，/scan トピックはロボットにつけられた Lidar センサなどのセン

表 6　想定システムにおけるトピック一覧 (抜粋)
Table 6　List of topics in the assumed system (excerpt)

topic	type	Hz
/battery_state	BatteryState	20
/cmd_vel	Twist	9.95
/constraint_list	MarkerArray	2
/imu	imu	20
/joint_states	JointState	20
/magnetic_field	MagneticField	20
/map	OccupancyGrid	1
/odom	Odometry	20
/scan	LaserScan	5
/scan_matched_points2	PointCloud2	5
/sensor_state	SensorState	20
/submap_list	SubmapList	3.35
/tf	TFMessage	238
/tf_static	TFMessage	1
/tf/_intra	IntraProcessMessage	20
/trajectory_node_list	MarkerArray	33.3

サ情報を扱うために使われるトピックであり，こちらも同様に障害物情報などをふくんでいることから，リスク値が高くなると考えられる

被害分析と攻撃分析をそれぞれ行った後，リスク値としてこれら 2 つのどちらのトピックも 5 と算出されたとする．このリスク値は 0 から 7 の 8 段階のリスクを想定し，最大リスク 7 とした場合の値としている．通常ではこのリスク値からセキュリティ対策として通信データの暗号化を実施することになるが，提案手法ではこの値に対し，リアルタイム性を考慮し図 6 の重みづけを行う．

/cmd_vel トピックは Twist 型（48 バイト）のデータを通信に使用し，通信周期は 10Hz であるため，図 6 に照らして，重みとして+1 を実行しリスク値としては 6 とする．

また，/scan トピックでは LaserScan 型を使用しており可変長の型ではあるが，我々の想定システムに搭載されている 2 次元 Lidar を使用した場合，データサイズは 3KB ほどとなる．周期は 5Hz であるため，これを図 6 に照らすと重みとして+2 となり, リスク値としてはこちらは最大値の 7 とする．もしロボットに必要なセンサが 3 次元のものなどより高性能なものに交換された場合など，/scan トピックによる通信データが 40KB を超えるような場合は，重みづけ表により重みは± 0 となり，リスク値として 5 を設定する．

このように算出リスクに対し重みづけを追加し，データサイズと周期に応じてリスク値を修正することで，暗号通信を行うかどうかの判定をする場合にリアルタイム性の考慮を含めることができる．

7. おわりに

ROS2 のセキュリティ機能のうち暗号機能について，その通信時間は，平文での通信に比べ 1.5 倍以上の時間がかかる．ロボットは実体として動作するため，通信速度の低下は動作に影響を与える可能性があるため，すべての通信を暗号化するのではなく，通信内容によって暗号化を選択する必要がある．そのためにはリスクに応じた選択をする手法が必要であり，算出したリスク値に対しリアルタイム性を含む手法を提案した．今後は事例への適用による有効性のさらなる確認と重みづけの妥当性の確認を行いたい．また，ロボットは用途に応じて様々な機能が追加され，ROS2 は機能追加を容易にするミドルウェアでもあることから，通信量，通信周期の増加による動的な暗号通信の変更についても検討したい．

参考文献

[1] 野村佳秀，木村功作，福寄雅洋，谷田英生：『オープンソースソフトウェア工学』シリーズ企業における OSS 活用の実際 (2016).

[2] Pangercic, D. and Sagar, M.: ISO 26262 Certification of ROS 2, *embedded world 2021* (2021).

[3] Maruyama, Y., Kato, S. and Azumi, T.: Exploring the Performance of ROS2, *Proceedings of the 13th International Conference on Embedded Software*, EMSOFT '16, New York, NY, USA, ACM, pp. 5:1–5:10 (online), DOI: 10.1145/2968478.2968502 (2016).

[4] Gerardo Pardo, R. W.: DDS Security concepts for SROS, `https://ruffsl.github.io/IROS2018\_SROS2\_Tutorial/content/slides/Background.pdf`.

[5] Esposito, C. and Ciampi, M.: On Security in Publish/Subscribe Services: A Survey, Vol. 17, No. 2, pp. 966–997.

[6] Belguith, S., Cui, S., Asghar, M. R. and Russello, G.: Secure Publish and Subscribe Systems with Efficient Revocation, *Proceedings of the 33rd Annual ACM Symposium on Applied Computing - SAC '18*, ACM Press, pp. 388–394.

[7] Shikfa, A., Önen, M. and Molva, R.: Privacy-Preserving Content-Based Publish/Subscribe Networks, *Emerging Challenges for Security, Privacy and Trust* (Gritzalis, D. and Lopez, J., eds.), Vol. 297, Springer Berlin Heidelberg, pp. 270–282.

[8] 亘理克好，大久保隆夫：ロボットオペレーティングシステムのセキュリティ機能に関する考察，情報処理学会研究報告 (2019).

[9] 亘理克好，大久保隆夫：ロボットオペレーティングシステムの対応 OS による暗号通信における性能評価，情報処理学会研究報告 (2019).

[10] Bonaci, T., Herron, J., Yusuf, T., Yan, J., Kohno, T. and Chizeck, H. J.: To Make a Robot Secure: An Experimental Analysis of Cyber Security Threats Against Teleoperated Surgical Robots (2015).

[11] Vilches, V. M., Gil-Uriarte, E., Ugarte, I. Z., Mendia, G. O., Pisón, R. I., Kirschgens, L. A., Calvo, A. B., Cordero, A. H., Apa, L. and Cerrudo, C.: Towards an open standard for assessing the severity of robot security vulnerabilities, the Robot Vulnerability Scoring System (RVSS) (2018).