1Z-5

自動車サービス連携のセキュリティアーキテクチャの提案

朝倉 知也† 岩井 大† 中道 上†‡ 青山 幹雄†‡
南山大学 数理情報学部 情報通信学科† 南山大学 情報理工学部 ソフトウェア工学科‡

1.研究の背景と課題

現在，車載システムと外部システム間の連携は，メーカ毎にインタフェースとプロトコルが異なるため，SOA (Service-Oriented Architecture)を適用し，SOAP や REST (REpresentational State Transfer)などの標準化されたプロトコルを用いるアーキテクチャが提案されている．しかし，このアーキテクチャでは外部システムとユーザ間のプロトコルが定義されていない．また，通信時のセキュリティが保証されていない．

2.関連研究

2.1. OSGi

JVM 上で動作するフレームワークであり，ソフトウェアコンポーネントである Bundle を連携可能である[5].

2.2. SOA に基づくアーキテクチャ

車載サービスブローカに OSGi を用いメッセージのプロトコル変換を行うことで，SOAP や REST を用いた連携ができる[2].

3.アプローチ

SOA に基づくアーキテクチャを拡張し，ユーザから車載システムまでの End-to-End でセキュアな通信を保証するアーキテクチャを提案する．前提条件として，外部システムのセキュリティは保証されているものとする．ユーザと外部システム間の通信は HTTPS (SSL)を用いる．また，外部システムと車載システム間の通信は SOAP(WS-Security)と REST(SSL)を用い，サービスの要求するセキュリティレベルに応じて選択可能とする(図 1).

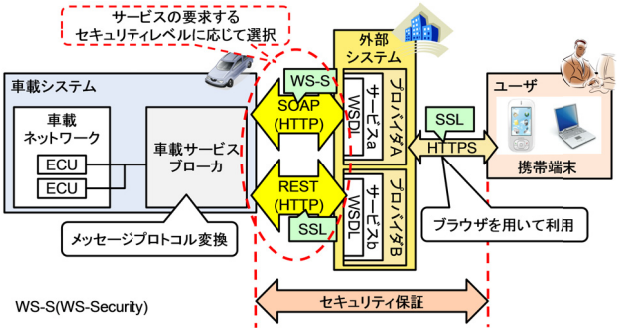


図 1 提案する連携方法

A Secure Architecture for Automotive Telematics Services
†Tomoya Asakura, Dai Iwai, Dep. of Information and Telecommunication Eng, Nanzan University.
‡Noboru Nakamichi, Mikio Aoyama, Dep. of Software Engineering, Nanzan University.

4.提案アーキテクチャ

4.1. セキュリティアーキテクチャ

車載サービスブローカに OSGi を用い，SOAP と REST を用いた連携を行う(図 2). OSGi フレームワーク上で SOAP/REST のプロトコル変換を行う Bundle が起動し，外部システムが選択したプロトコルの通信に対応することで，外部システムと車載ネットワークの連携を可能とする．ユーザと外部システム間，外部システムと車載システム間で WS-Security または SSL を用いる．暗号化通信とユーザ認証を行うことで，End-to-End でセキュアな通信を保証する.

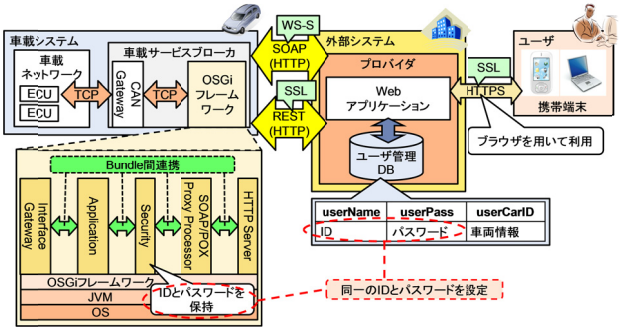


図 2 セキュリティアーキテクチャ

4.2. セキュリティの選択

外部システムと車載システム間の通信には，SOAP(WS-Security)と REST(SSL)を選択可能である．外部システムの要求するセキュリティレベルに応じ，WS-Security と SSL を選択する(図 3).

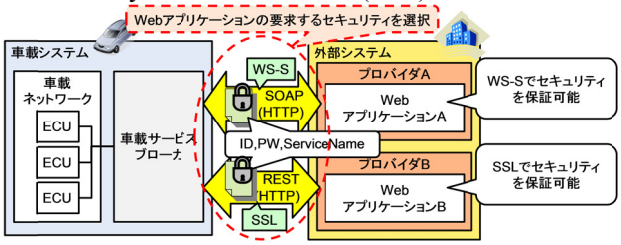


図 3 セキュリティの選択

5.提案アーキテクチャのプロトタイプ

5.1. 利用シナリオ

プロトタイプの利用シナリオとして，遠隔からドアのロック状態を確認する「ドアロック確認システム」を選択する．前提条件として，このシステムが要求するセキュリティレベルは SSL で満たせるものとする．また，比較対象としてセキュリティ機能の無いドアロック確認システムを実装し，応答時間を計測する.

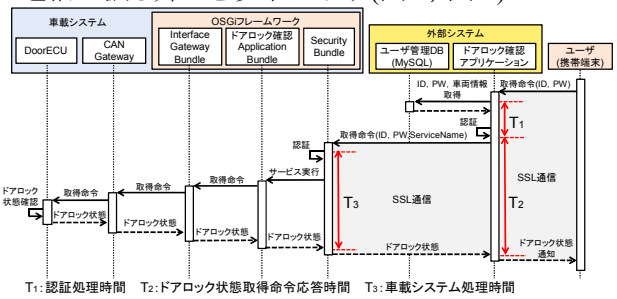
5.2. 実装環境

車載サービスブローカのプロトタイプでは，OSGi フレームワークとして Knopflerfish[3]を用いる．また，外

部システムに Apache Tomcat[1], ユーザ管理 DB に MySQL[4]を用いる. プロトタイプは Java で実装し, 規模は外部システムが 108 行, 車載システムが 216 行となった.

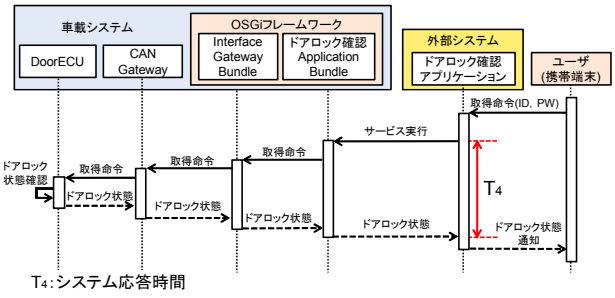
5.3. プロトタイプの振る舞い

セキュリティ機能が有る通信とセキュリティ機能が無い通信の振る舞いを以下に示す(図 4, 図 5).



T1:認証処理時間　T2:ドアロック状態取得命令応答時間　T3:車載システム処理時間

図 4 セキュリティ機能が有る通信のシーケンス図



T4:システム応答時間

図 5 セキュリティ機能が無い通信のシーケンス図

6.プロトタイプに基づく評価

6.1. プロトタイプの性能評価

実装したドアロック確認アプリケーションの起動から終了までの応答時間を測定し, 性能を評価した. 実装した DoorECU と CAN Gateway はスタブであり, 車載システムの正式な応答時間を測定することはできないため, セキュリティ機能の処理時間に着目する. セキュリティ機能が無い通信と比較し, セキュリティ機能の処理時間を測定する.

6.2. 応答時間の測定と平均値の算出

セキュリティ機能が有る通信とセキュリティ機能が無い通信の応答時間を比較するためそれぞれ 1000 回実行し, 図 4 と図 5 で示した T1+T2 と T4 の測定を行った. JVM によるガベージコレクションによる遅延を特異値とし, 特異値を除いた補修値を算出した(図 6).

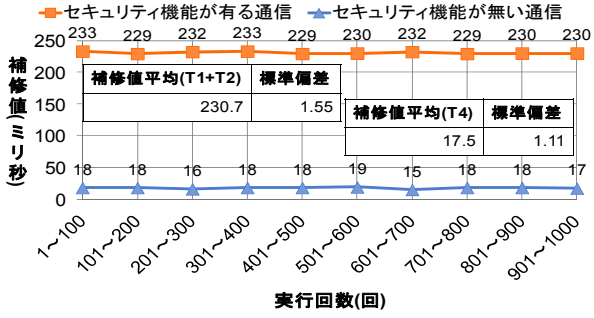


図 6 100 回ごとの補修値平均

6.3. セキュリティ機能の処理時間

セキュリティ機能が有る通信とセキュリティ機能が無い通信の応答時間の差が, セキュリティ機能の処理時間である. また, 図 4 で示した T1 がユーザ認証, T2-T3 が SSL 通信の処理時間である(表 1). この結果より, SSL 通信処理が応答時間の大部分を占めることを確認した.

表 1 応答時間とセキュリティ機能の処理時間

	補修値平均(ミリ秒)
セキュリティ機能の有る通信(T1+T2)	230.7
セキュリティ機能の無い通信(T4)	17.5
セキュリティ機能の処理時間	213.2

	平均応答時間(ミリ秒)
認証処理時間(T1)	30.4
外部システムと車載システム間のSSL通信処理時間(T2-T3)	175.2

7.考察

7.1. セキュアな通信

SOA に基づく車載サービスブローカアーキテクチャを拡張し, 車載システムからユーザまでのセキュアな通信を実現した. また, 外部システムが要求するセキュリティレベルに応じ, WS-Security と SSL から適切なセキュリティを選択可能とした.

7.2. プロトタイプの応答時間

セキュリティ機能が有る通信とセキュリティ機能が無い通信を比較し, セキュリティ機能の処理時間を測定した. 表 1 のセキュリティ機能の処理時間と, ユーザ認証と SSL 通信の処理時間の和がほぼ等しいことから, 双方の測定結果が妥当であることが確認できる. その結果, SSL 通信がセキュリティ機能処理時間の約 8 割となることを確認した. これは, Java では SSL 通信の暗号処理に時間がかかるためだと考えられる. この問題の解決方法として, JNI(Java Native Interface)を用いた SSL 通信処理のパフォーマンス改善がある.

8.今後の課題

(1) SSL 通信処理のパフォーマンス改善
(2) セキュリティ選択方法のコード化
(3) WS-Security を用いたプロトタイプの実装
(4) 正式な ECU と CAN Gateway の利用

9.まとめ

本稿では End-to-End でセキュアな通信を保証するアーキテクチャを提案した. 異なるセキュリティレベルの要求に応じ, WS-Security と SSL を選択可能とした. また, プロトタイプを開発し, 評価を行った.

参考文献

[1] Apache Tomcat, http://tomcat.apache.org/.
[2] 濱千代 正弥, 片桐 雅仁, 自動車ネットワークサービスの連携アーキテクチャ 南山大学 2010 年度卒業論文, 2011.
[3] Knopflerfish OSGi-Open Source OSGi Service Platform, http://www.knopflerfish.org/index/html/.
[4] MySQL, http://www-jp.mysql.com/.
[5] OSGi(Open Service Gateway initiative) Alliance, http://www.osgi.org/Main/HomePage.