7ZC-05

組み込み機器向け制御フローアテステーションで用いるホワイトリスト検証手法の検討

高石 颯汰* 福田 洋治* 廣友 雅徳† 白石 義明‡

近畿大学* 佐賀大学† 神戸大学‡

1. はじめに

近年，IoT 機器は様々な分野での導入が進められており，IoT セキュリティに関する懸念が緊急性を増している．しかし，既存の組み込みソフトウェア向けのセキュリティは，ほとんどのものがアプリケーションの制御やデータのフローを操作するランタイム攻撃を検知できない．

IoT 機器上で動作するプログラムに対する実行パスに影響を与える攻撃を検知するために，対象プログラムのモジュールに対して CFG（Control Flow Graph）をもとにブロックに分割，ブロック毎に実行パスの auth 値を計算，使用するリモートアテステーション手法(C-FLAT)[1]が提案されている．

これまで，著者らは，C-FLAT の文献ではホワイトリストの構成，検証について詳細が言及されていないことから，C-FLAT のホワイトリストの構成法を検討し，ソースコードの CFG を得た後で深さ優先探索により各実行パスの auth 値を計算しホワイトリストを構成する手法を提案している[2]．

本稿では，C-FLAT 向けのホワイトリストの構成，検証について継続して検討し，当該実行パスの auth 値を計算，突き合わせる，事前にホワイトリストを構成しない動的な手法を提案し，事前にホワイトリストを構成する場合との計算量とデータサイズを比較し，考察を述べる．

2. 関連研究

小松らは，大きな被認証プロセスに対して脆弱性が想定される部分に対してのみ実行パスの累積ハッシュ値の計算を適用し，実行速度の低下を最低限に抑えつつ効率的な認証を行う手法を提案している[3]．

Ahmed らは，静的認証と動的認証の二つの認証フェーズをもつリモートアテステーション手法 SAPEM を提案しており，動的認証フェーズではループのハッシュ計算と条件分岐の管理に新たなアプローチを導入することで認証時のオーバーヘッドを抑えている[4]．

3. C-FLAT について

C-FLAT は，バッファオーバーフローによるコードインジェクションや，コード再利用攻撃のような，プログラムの制御フローを変更する攻撃を検知するために，ターゲットプログラムを任意の分岐（直接分岐，間接分岐，ジャンプ，コール，リターンなど）ごとに基本ブロックに分割し，ブロック毎に実行パスのハッシュ値を計算，使用することでリモートアテステーションを行う手法である．

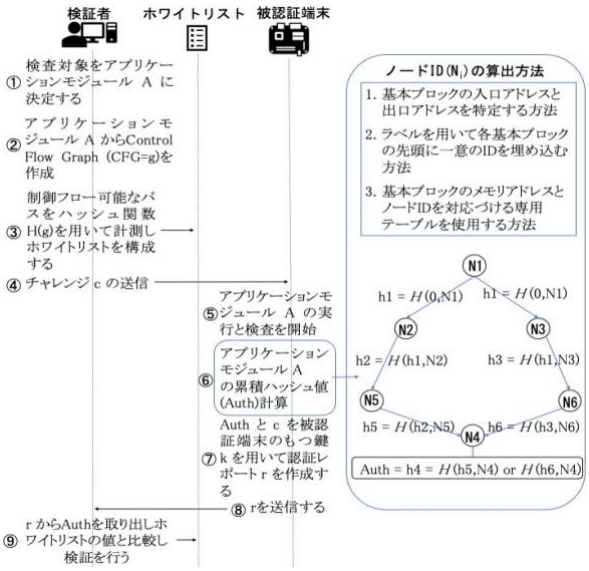

図 1 C-FLAT のリモートアテステーションの例

C-FLAT のリモートアテステーションの流れを図 1 に示す．

①認証の対象のアプリケーションモジュール A を決める．

②アプリケーションモジュール A を静的解析し CFG を作成する．

③全実行パスの計測値からホワイトリストを構成する．

④検証者は，被認証端末にアプリケーションモジュールの名前とランダムな値を含むチャレンジ c を送る．

⑤被認証端末は，チャレンジ c を受け取るとアプリケーションモジュール A の実行を開始する．

⑥アプリケーションモジュール A の実行に伴うハッシュ値を計算する．次のノードに移行する時に直前のハッシュ値 H_{prev} とノードを一意に識別するノード ID N_i を用い $H_i=H(H_{prev}, N_i)$ を計算する．

⑦被認証端末がもつ鍵 k を使用してチャレンジ c と Auth を暗号化し，認証レポート r を作成する．

⑧認証レポート r を検証者に送信する．

⑨検証者は認証レポート r を受け取ると，認証レポート r の Auth とホワイトリストの値を比較し検証を行う．

4. 事前にホワイトリストを構成しないリモートアテステーション手法の検討

C-FLAT のホワイトリストは，事前準備で作成したターゲットプログラムの CFG を，深さ優先探索によって全実行パスを求め，各実行経路についてハッシュ計算を行い，得た認証値から構成することができる．検証時には上記の方法で構成した，ホワイトリストを全探索することで，被認証端末から得る Auth を検証する．

A Method of White List Verification in Control Flow Attestation for Embedded Devices

* Sota Takaishi and Yoji Fukuta・Kindai University

† Masanori Hirotomo・Saga University

‡ Yoshiaki Shiraishi・Kobe University

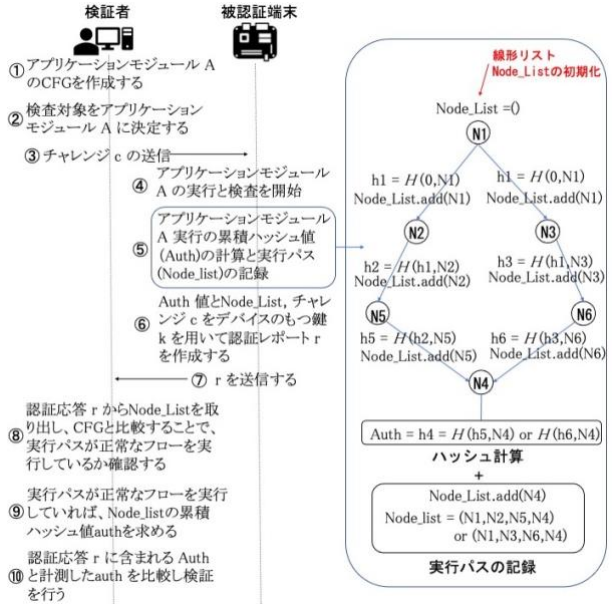

図 2 提案手法のリモートアテステーションの例

しかし，ターゲットプログラムが複雑で大規模になるほど，実行パスの個数が増加し，検証時の計算量が大きくなる．

そこで事前にホワイトリストの構成をせずに，被認証端末から認証値と実行パスを得ることで当該実行パスの Auth を計算，突き合わせる動的な手法を提案する．

提案手法のリモートアテステーションの流れを図 2 に示す．

①認証の対象のアプリケーションモジュール A を決める．

②アプリケーションモジュール A を静的解析し CFG を作成する．

③検証者は，被認証端末にアプリケーションモジュールの名前とランダムな値を含むチャレンジ c を送る．

④被認証端末は，チャレンジ c を受け取るとアプリケーションモジュール A の実行を開始する．

⑤アプリケーションモジュール A の実行パスの累積ハッシュ値 Auth と，実行パスとしてハッシュ計算に使用するノード ID のリストを Node_List に記録する．

⑥被認証端末がもつ鍵 k を使用してチャレンジ c と Auth, Node_List を暗号化し，認証レポート r を作成する．

⑦認証レポート r を検証者に送信する．

⑧検証者は認証レポート r を受け取ると，認証レポート r の Node_List と CFG を比較し被認証端末が正常な制御フローを実行していることを確かめる．

⑨正常な制御フローを実行している場合，Node_List に対して累積ハッシュを計算し auth を求める．

⑩認証レポート r の Auth と検証者が求めた auth を比較することで検証を行う．

手順⑧で，被認証端末が正常な制御フローを実行していることが確認できなかった場合，それ以降の手順は行わず，検証者にターゲットプログラムが不正な実行が行われたことを通知する．

5. 計算量とデータサイズの比較と考察

事前にホワイトリストを構成するリモートアテステーション手法と提案手法の事前準備と検証時にかかる計算量とデータサイズをそれぞれ比較する．計算量とデータサイズは，CFG の代わりに深さn分岐mの完全m分木を仮定して求める．

表 1 に，一つの値に必要なデータのサイズを αSize，ハッシュ値に必要なデータサイズを AuthSize とおいて求めた，事前準備と検証時にかかる計算量とデータサイズを示す．

表 1 計算量とデータサイズ

		ホワイトリストを用いた手法	提案手法
計算量 [Times]	事前準備	$(n+3)m^n$	m^n
	検証時	m^n	$2n+2$
データサイズ [byte]	事前準備	$((\sum_{k=1}^{n} m^k)+2)(m+1)$ $\cdot \alpha Size + m^n \cdot AuthSize$	$((\sum_{k=1}^{n} m^k)+2)(m+1)$ $\cdot \alpha Size$
	検証時	$(m^n+1) \cdot AuthSize$	$\left(((\sum_{k=1}^{n} m^k)+2)(m+1)+(n+1)\right) \cdot \alpha Size + AuthSize$

事前にホワイトリストを構成する手法と提案手法では，事前準備と検証時のどちらも提案手法による処理の計算量が少なくなっている．また，提案手法による検証は，ターゲットプログラムの分岐回数によって計算量が増減しないため，ターゲットプログラムが複雑で大規模なプログラムの場合でも，検証にかかる計算量の増加を抑えることができると考えられる．

また，事前準備と検証時の必要となるデータサイズでは，事前にホワイトリストを構成する手法の事前準備に必要となるデータサイズが最も大きい．その為，事前にホワイトリストを構成する手法に比べ，提案手法が必要とするメモリの最大使用量が少なくなっていると考えられる．

しかし，検証時に必要となるデータサイズは，事前にホワイトリストを構成する手法に比べ，提案手法の方が大きい為，検証時に使用できるメモリ容量が限られている場合，本提案手法を適用できない可能性があると考えられる．

6. まとめ

事前にホワイトリストを構成し C-FLAT でリモートアテステーションを行う場合，ターゲットプログラムが複雑で大規模になるほど，検証時の計算量が大きくなることから，事前にホワイトリストを構成しないリモートアテステーション手法を提案した．また，提案手法と事前にホワイトリストを構成する手法の計算量とデータサイズを算出し，比較することで考察を行った．

今後の課題として，検証時のデータサイズが大きくなってしまうこと，実行パスのぶんだけ通信量が増えてしまうことから，CFG のデータサイズの削減，検証者に送信するデータの削減の検討が挙げられる．

参考文献

1) Tigist Abera, N. Asokan, Lucas Davi, Jan-Erik Ekberg, Thomas Nyman et al.: C-FLAT: Control-Flow Attestation for Embedded Systems Software, The 2016 ACM SIGSAC Conference on Computer and Comm. Security, pp.743-754 (2016).

2) 西本拓矢，福田洋治，廣友雅徳，白石義明: IoT 機器上で動作するプログラムの改ざん検知で用いるホワイトリストの作成方法の検討，電子情報通信学会 2021 年総合大会予稿集，A-7-8 (2021).

3) 小松昌平，PEYROUTAT-BASSE Jerome，瀧本栄二，毛利公一，斎藤彰一:プロセス処理内に対する遠隔認証手法の提案と実装，CSS2018 予稿集，pp.22-25 (2018)

4) Nafisa Ahmed, Manar Abu Talib, Qassim Nair: SAPEM: Secure Attestation of Program Execution and Program Memory for IoT Applications, Computers, Materials & Continua, vol.67, no.1, pp.23-49 (2021).