JTAG 認証機構の軽量化設計について

On the Low-Cost design of JTAG Authentication Mechanism

馬　竣[†]、岡本　悠[†]、王　森レイ[†]、甲斐　博[†]、亀山　修一[†]、高橋　寛[†]、清水　明宏[††]

[†]愛媛大学大学院理工学研究科
[††]高知工科大学

1.はじめに

近年の IoT（Internet of Things）技術の急速な普及に伴い，膨大な数のデバイスがインターネットに接続して動作している。これらの IoT デバイスは，コスト要因で十分なセキュリティ対策が施されていないため，サイバー攻撃の標的となる恐れがある [1]。

IoT デバイスに対する攻撃では，デバイスに組込まれる JTAG ポートが攻撃の糸口になる恐れがある。JTAG とは，集積回路(IC)や基板の動作テストを行うための国際標準規格（IEEE 1149.1）である。JTAG は少数ピンで IC の内部回路を容易にアクセスできるため，出荷後のデバイスに対するデバッグやファームウェア更新などにも広く利用されている [2]。JTAG にセキュリティ対策を講じないと，悪意のある攻撃者が JTAG を介してデバイスを乗っ取り，インターネットを介して IoT システムに対する攻撃を行う恐れがある。

JTAG のセキュリティ強化のためには，JTAG 機構に暗号技術を用いた認証回路を組込むことで，正しく認証できるユーザーのみに対してテストアクセスポート(TAP)へアクセスを許可するアクセス制御方法が提案されている。しかしながら，これらの認証機能を実現するために，IC 製品に暗号回路を含む専用なハードウェアを追加することが必要であるため，コスト重視の IoT デバイスに導入することは困難である [3]。

本研究では，JTAG 認証の軽量化技術の開発を目的とする。本稿では，SAS というワンタイムパスワード認証方式を用いた JTAG 認証プロトコルを提案する。提案法では，認証処理のほとんどがサーバー側で行い，クライアント側の演算負荷が小さいため，より少ないハードウェアでの JTAG 認証機能を実現することが可能となる。

本稿の構成は以下の通りである。2 章では，従来の JTAG 認証機構とその問題点について述べる。3 章では，SAS というワンタイムパスワード認証方式の基本を説明する。4 章では，SAS を用いた JTAG 認証プロトコルを提案する。5 章では，まとめと今後の課題について説明する。

2. JTAG アクセス認証

2.1 従来の認証方法 [3]

JTAG への不正アクセスを防ぐために，Novak らは [4]，JTAG インフラストラクチャーにロッキング回路を追加することで TAP コントローラーを制御するアクセス認証機構を初めて提案した。図 1 は，認証機構のイメージを示す。TAP は通常ロッキング回路によってロックされている状態で，ユーザーは JTAG ポートをアクセスする際に，パスワードの入力が求められ，デバイスに格納されている認証データと照合し，一致した場合 TAP が解除され，すべての JTAG インフラストラクチャーへのアクセスを許可する。また，盗聴などによるパスワードの漏洩を防ぐためには，共通鍵暗号や公開鍵暗号などの暗号回路を導入した認証機構も提案されている [5][6][7]。

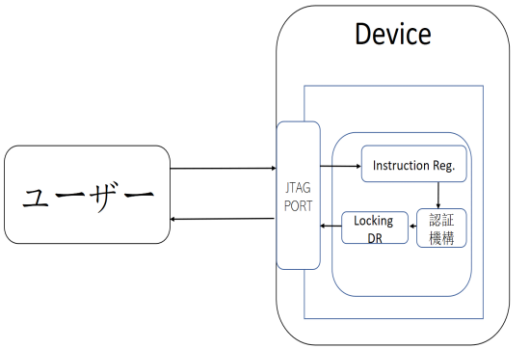

図 1．従来の JTAG 認証

2.2 従来の認証方法の課題

課題１：ハードウェアコストが高い

従来の JTAG アクセス認証は，強力な暗号化方式を導入すれば，十分なセキュリティを確保することができると考えられる。一方，デバイス側に暗号回路を含む専用なハードウェアリソースを追加することが必要である。IoT システムにおいては，コスト要因で低スペック製品がほとんどであるため，コストのかかる認証専用のハードウェアを実装することが困難である。

課題２：セキュリティの脆弱性

IoT システムにおいては，デバイスが現場に置かれ，デバイス間で頻繁にデータのやり取りを行っている。盗聴による大量なデータを簡単に獲得することが可能であるため，総当たり攻撃が容易に実施することができる。従来の認証方法では，秘密鍵などの認証データはデバイス側に格納し，固定されている。暗号化された認証データが盗聴される場合，総当たり攻撃による秘密鍵が割り出されるリスクがある。

3. SAS による JTAG 認証

コストを重視する IoT デバイスのセキュリティを強化するために，軽量な認証方式が求められる。そこで，本研究では， SAS というワンタイムパスワード認証方式を用いた JTAG 認証プロトコルを提案する。

3.1 SAS ワンタイム認証 [5]

SAS(Simple And Secure password authentication protocol)とは、高知工科大学の清水明宏教授が考案したワンタイムパスワード認証方式である。他の認証方式と比較し，被認証側の演算負荷が極めて低いため，コストを重視する処理能力の低い IoT デバイスに

おいて軽量かつ強固なワンタイム認証を実現することが期待されている。

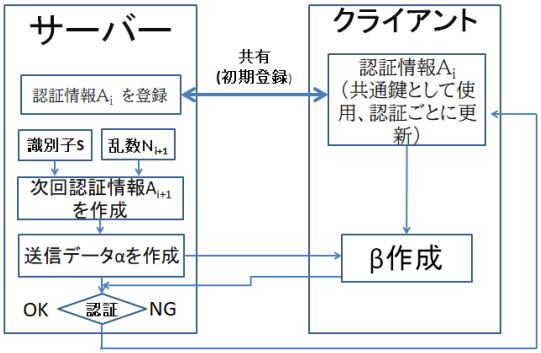

図２．SAS 認証手順

SAS では主に初回認証の前に行われる登録フェーズと認証フェーズに分かれており，図２に示す手順で認証（サーバー側）⇔被認証側（クライアント）間の相互認証を実現する。

登録フェーズ：

(ア) ユーザーはサーバー側で乱数を用いて識別子 S（ID とパスワードなど）を暗号化し，初期認証情報 A_i を作成する。

(イ) 初期認証情報 A_i を安全なルートでクライアントと共有する。

認証フェーズ：

(ア) サーバー側で新しい乱数 N_{i+1} を生成し，識別子（S）を暗号化し，次回認証情報 A_{i+1} を作成する。

(イ) サーバー側で送信データ α を以下の式で作成し，インターネットを介してクライアント側に送信する。

$$\alpha = A_i \oplus A_{i+1} \qquad \text{式１}$$

(ウ) クライアント側では格納されている前回の認証データ A_i を α と排他的論理和演算を取ることで，サーバー側で生成された現在の認証情報 A_{i+1} を復号し，一時保存する。

(エ) クライアント側で，以下の式で送信データ β を作成し，サーバーに送信する。

$$\beta = \alpha \oplus A_i + A_i \qquad \text{式２}$$

(オ) サーバー側で $A_i + A_{i+1}$ の演算を行い，$A_i + A_{i+1}$ と β が等しい場合は認証成立となる。

(カ) 認証が成功した場合，クライアント側に認証成功のメッセージを送信し，サーバー側とクライアント側の認証データ A_i を A_{i+1} に更新する。

(キ) 認証が失敗した場合，クライアント側に認証失敗のメッセージを送信し，サーバー側とクライアント側の認証データ A_i を A_{i+1} に更新しない。

SAS の認証手順により，サーバーで生成した次回認証情報とあらかじめクライアントとサーバーが所持していた今回の認証情報との排他的論理和演算の結果を配送することで，識別情報（S）をネットワークに直接流すことなく認証および鍵の更新を行うことができるため，総当たり攻撃に強い。さらに，演算負荷の大きいワンタイム認証情報作成及び乱数の生成はサーバーが担い，クライアント側では暗号化された認証情報を抽出するための排他的論理和演算と加算の簡単な演算のみとなっているため，ユーザーの処理負荷が極めて小さくなる。

3.2 SAS を用いた JTAG 認証プロトコル

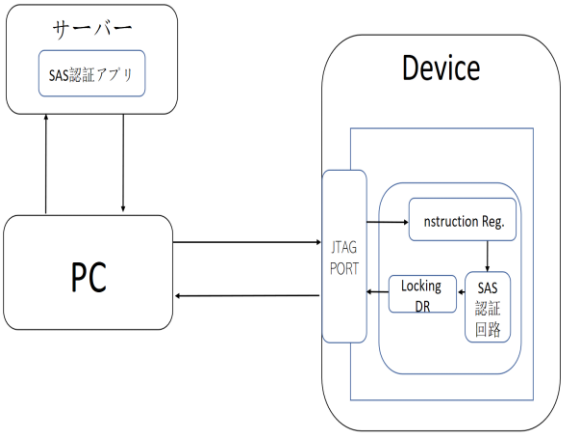

図 3. ＳＡＳによる JTAG 認証

JTAG における SAS 認証方式の実現には，IC 製品の出荷前にメーカーによるデバイスとユーザーの識別情報（デバイス ID とパスワードなど）を暗号化し，初期認証情報としてデバイスのメモリに書き込むことが必要である。また，IC 製品ごとのデバイス識別情報とユーザー情報を管理するために，メーカー側はデバイス管理サーバー（DMS）を用意することが必要である。図 3 に SAS による JTAG アクセス認証のイメージを示す。

IC 製品の出荷後，ユーザーはオンチップデバッグ(OCD)ツールを用いて，以下の手順でデバイスに組込まれる JTAG にアクセスする。

Step 1: ユーザーが OCD ツールで製品メーカーのデバイス管理サーバーにアクセスする。アクセスする際に別途の認証が求められる。

Step 2: ユーザーが DMS サーバーにおいてターゲットデバイスへのアクセス要請を出す。

Step 3: DMS サーバーがユーザーからのアクセス要請を受理し，当該デバイスの識別情報に対して，認証データ A_{i+1} を新規に生成し，式１で認証コード α を作成してユーザーに配布する。

Step 4: ユーザーが DMS から受け取った認証コード α を OCD ツールを用いて JTAG に送信する。

Step 5: JTAG デバイス側では，ユーザーが入力した認証コード α に対して，セキュアメモリに格納されている前回認証データ A_i と排他的論理和を取ることで，DMS 側で生成した新規認証データ A_{i+1} を解く。

Step 6: JTAG デバイスで，A_i と A_{i+1} 及び α を用いて，式２で認証データ β を生成し，ユーザーにフィードバックする。

Step 7: ユーザーが β を DMS にアップロードして，認証データの照合処理を行う。

Step 8: DMS が認証成立の場合は $A_i + A_{i+1}$ を，認証失敗の場合は乱数を認証結果としてユーザーに発行する。

Step 9: ユーザーが DMS から受け取った認証結果を JTAG に送信する。

Step 10: JTAG では，認証結果を $A_i + A_{i+1}$ と比較し，一致した場合セキュアメモリのデータを A_{i+1} に更

新し，アクセス権限を開放する。一致でない場合，A_i を更新せず，TAP をロックしたままにする。

以上のプロトコルは，JTAG 対応製品に強力なセキュリティ対策をより低コストで講じることが可能であると考えられる。その理由としては以下の点が挙げられる。

コストメリット：

1. デバイス側に排他的論理和演算と加算などの簡単な演算回路のみが必要のため，認証用ハードウェアが極めて少ない。
2. ワンタイムパスワード認証に関わる計算（乱数と一方性関数など）は DMS サーバーに集中するため，よりコストの少ないソフトウェアで大量なデバイスに対して効率的に認証を行うことが可能である。デバイスの数が多ければ多いほど，コストメリットが大きい。

セキュリティメリット：

3. ワンタイムパスワード認証の実現による総当たり攻撃に強い。特にデバイス間で頻繁にデータのやり取りを行っている IoT システムにおいて，そのメリットが大きい。
4. ワンタイムパスワード認証に必要な認証情報の生成は DMS サーバーで行い，必要に応じて生成アルゴリズム（乱数・一方性関数）をアップグレードすることができるため，セキュリティが高い。
5. DMS サーバーに登録されているユーザー情報と製品データを保護するために，多要素認証などの強力な認証方法を簡易に導入することができる。

4.まとめ

コスト重視の IoT デバイスにおける JTAG のセキュリティを強化するために，クライアント側の演算負荷の少ない SAS ワンタイムパスワード認証方式を用いた JTAG 認証プロトコルを提案した。今後は，FPGA において，提案した認証メカニズムのハードウェア設計と実装評価を行う予定となる。

参考文献

1) エッジが攻撃対象を拡大する懸念,https://cafe-dc.com/special/edge-increases-attack-surface/,accessed on 3 Jan 2022.

2) ケネス・Ｐ．パーカー(著)，亀山修一(監訳)：“バウ ンダリスキャンハンドブック，”青山社，2012 年，第 3 版，ISBN978-4-88359-303-3

3) 王 森レイ，亀山 修一，高橋 寛,“JTAG のセキュリティ脅威 ―攻撃の現状とその対策―, ”エレクトロニクス実装学会誌,Vol.24, No.7, pp.668- 674, 2021.

4) F. Novak and A. Biasizzo, “Security extension for IEEE Std 1149.1,” J. Electron. Test. Theory Appl., vol. 22, no. 3, pp. 301–303, 2006.

5) G. Chiu and J. C. Li, "A Secure Test Wrapper Design Against Internal and Boundary Scan Attacks for Embedded Cores," in IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 20, no. 1, pp. 126-134, Jan. 2012, doi: 10.1109/TVLSI.2010.2089071.

6) Amitabh Das, Jean da Rolt, Santosh Ghosh, Stefaan Seys, Sophie Dupuis, et al. “Secure JTAG Implementation Using Schnorr Protocol,” J. Electron. Test. Springer Verlag, 2013, 29 (2), pp.193-209. 10.1007/s10836-013-5369-9.

7) R. F. Buskey and B. B. Frosik, "Protected JTAG," 2006 International Conference on Parallel Processing Workshops (ICPPW'06), 2006, pp. 8 pp.-414, doi: 10.1109/ICPPW.2006.65.

8) SAS-L2,https://www.kochitech.ac.jp/power/research/sas-liot.html, accessed on 3 Jan 2022.