参考資料

Application Note

TI バッテリー残量計への楕円曲線暗号による認証機能の実装

Michael Smith, Garry Elder

概要

TI のほとんどのバッテリ残量計は、バッテリ パックの偽造防止保護として、いくつかのレベルの認証機能をサポートしています。このアプリケーション ノートは、認証機能を要約し、BQ41zxx 製品ファミリを例として、楕円曲線暗号 (ECC) キーペアの作成およびプログラミング プロセスの詳細を示します。

目次

商標

1 はじめに

楕円曲線暗号化 (ECC) は、楕円曲線の数学的性質を利用して非対称のプライベートキーと公開キーのペアを生成する認証方式です。ECC アルゴリズムは、ECDSA (FIPS 186-5 で詳述) や EC-KCDSA などのいくつかの異なるバージョンで利用可能です。TI のバッテリ残量計である BQ41Zxx ファミリは、KCDSA タスクフォース チームが公開した論文に基づいて実装された、韓国の認証に基づく楕円曲線デジタル署名アルゴリズム (EC-KCDSA) を使用しています。

BQ41zxx デバイスファミリで使用されている実装は、B-233 に基づいて EC-KCDSA シグネチャ (つまりチャレンジからの応答) を提供し、ハッシュとして SHA-256 アルゴリズムを使用します (FIPS 183-4 で詳細を説明しています)。実装では、公開キーの X 座標と Y 座標を使用し、適切な長さにパディングされます。

ゲージの認証機能は、BQ41z50 テクニカル リファレンス マニュアルに詳細を示された MaufacturerAccess( ) コマンドを使用して、SMBus インターフェイス経由でアクセスします。BQ41z50 は、ノート PC などのホスト デバイスで認証でき、ゲージは、再構成または再プログラムできるようにホストを認証することもできます。

表 1-1. 使用された SMBus コマンドの概要

タイプ	ID	機能	モード	アクセス権
MAC	0x0034	HostPublicKey( ) ホスト認証の公開キーの読み取りと書き込みを許可します。 注 1 - ホスト認証の公開キーが設定されると、従来の「2 ワードの UNSEAL」方式はただちに無効になります。 注 2 -フルアクセス モードでホスト認証を無効にするには、これをすべて 0 に書き戻すことができます。	読み出し / 書き込み	R: S/U/F W: F
MAC	0x0036	GaugeAuthPubKey( ) デバイスを認証するための単一の「圧縮ポイント」公開キー読み出しは、キーステータスバイトと公開キーの 30 バイトを返します (LSB ファーストで圧縮)	読み出し	S/U/F
MAC	0x0038	ProdPrivateKey( ) ゲージ認証秘密キー (秘密キー 30 バイト + 公開キー圧縮ポイント 30 バイト) のプログラムに使用	書き込み専用	F
MAC	0x003a	ECC_MAC( ) ホストが認証済み UNSEAL コマンドを実行するために使用されます	読み出し / 書き込み	S/U/F
MAC	0x003c	ECC_R 読み出しは、利用可能であれば最新のゲージ認証結果 *r* を返します。 書き込みは、ホストが ECC_MAC() の一部として認証データをゲージに書き込むためのものです。	読み出し / 書き込み	S/U/F
MAC	0x003d	ECC_S 読み出しでは、利用可能な場合、ゲージの認証結果 *s* が返されます 書き込みは、ホストが ECC_MAC() の一部として認証データをゲージに書き込むためのものです。	読み出し / 書き込み	S/U/F
SBS	0x2f	GaugeAuthentication( ) *r* と *s* の 60 バイトを測定し、読み出すためのチャレンジを書き込むために使用されます。	読み出し / 書き込み	S/U/F

規格の詳細については、セクション 8 を参照してください。

2 認証方式の比較

ECC には、SHA-1 などのセキュアハッシュアルゴリズム (SHA) 認証方式と比較していくつかの重要な違いがあります。第 1 に、ECC は非対称キーを使用するため、ホストとゲージは 1 つのキーも秘密も共有しません。また、デバイスの認証にキーペア (1 つの公開キーと 1 つのプライベートキー) を使用する必要があります。どちらの方法も、ランダム チャレンジを提供するために 20 バイトのチャレンジ長を持つことができます。

2 つの認証プロトコルの大きな違いの 1 つは、認証の検証に関するものです。SHA を使用すると、ホストはゲージと並行して検証を開始できます。これは、ホストには検証が開始された時点でプライベートキーとランダムチャレンジが格納されており、一方、ゲージがチャレンジを受信するとゲージが開始されます。ECC では、ホストがゲージからの応答を待ち、検証プロセスを完了する必要があります。ただし、ホストはこの応答を待っている間、特定の計算を事前に開始できます。

2 つの認証プロトコルの 2 つ目の大きな違いは、キープログラミングの検証です。同じキーとチャレンジを使用すると、SHA は同じ応答を生成します。したがって、チャレンジと応答のペアを使用して、キーが正しくプログラミングされているかどうかを確認できます。ECC では、同じキーと同じチャレンジが同じ応答を生成しません。キーが正しくプログラムされていることを確認するには、別の検証機能を実装する必要があります。

表 2-1. ECC v. SHA-1 認証アルゴリズム

	SHA-1 HMAC	ECC
TI 製品	BQ40z50、BQ41z50	BQ41z50
キータイプ	対称キー (秘密を共有)	非対称キー (公開キーとプライベートキーのペア)
ハッシュ関数	160 ビット	256 ビット (SHA-2)
キー長	128 ビット	233 ビット キー
認証応答時間	<100ms	<100ms
チャレンジの長さ	20 バイト	8 - 19 バイト
応答の長さ	20 バイト	60 バイト (または 2 x 30 バイト)
所与のキーとチャレンジに対する確定的な応答	なし	あり
プライベートデータを使用せずにプログラミングされたキーを検証	あり 既知のチャレンジペア/応を使用	あり 公開キーを使用して機能を検証する

3 パック製造フロー内での ECC キーのプログラミング

BQ41z50 には、表 3-1 に示されているように、目的を変更するためのいくつかのキーのセットが含まれています。

表 3-1. キーのペアと秘密のリスト

キー	所有者	Use
ホストコマンドキー (公開専用)	顧客	ホストコマンドの認証に使用
ゲージ認証キー (プライベート、公開)	顧客	BQ41z50 デバイス向けに生成およびプログラミングされたホスト。 バッテリホストシステムの認証に使用。

ゲージとホストの生産キーのペアは、生成するのに特定の手順を実行し、BQ41z50 にアップロードする必要があります。

キーの生成:プライベートキーのランダム 233 ビット値を「x」として、公開キーを (x^-1)G として生成するか、または標準 ECDSA キーを生成し、(ECKCDSA プライベートキー) = (ECDSA プライベートキー)^-1 を生成します

キーのプログラミング (フルアクセスのみ):プライベートキー (30 バイト、LSB ファースト) || 公開キー (30 バイト、LSB ファースト、圧縮)を MAC 0x0038 に書き込みます

4 BQ41z50 製品ファミリのゲージ認証フロー

認証プロセスは、パックメーカーまたは組み立て業者が行い、完全な認証済みバッテリパックの製造を保証します。認証フローは、ゲージに SMBus ManufacturerAccess() (MAC) コマンドを使用して、図 4-1 図に視覚化されます。

フローの概要は以下のとおりです。

1. ホストがゲージから ProdKpub を読み取ります
2. ゲージの認証
   - ホストがゲージにチャレンジメッセージを送信します
   - ゲージは ECC-233 でメッセージに署名します
   - ゲージはホストに r/s を提供します
   - ホストが署名を検証します

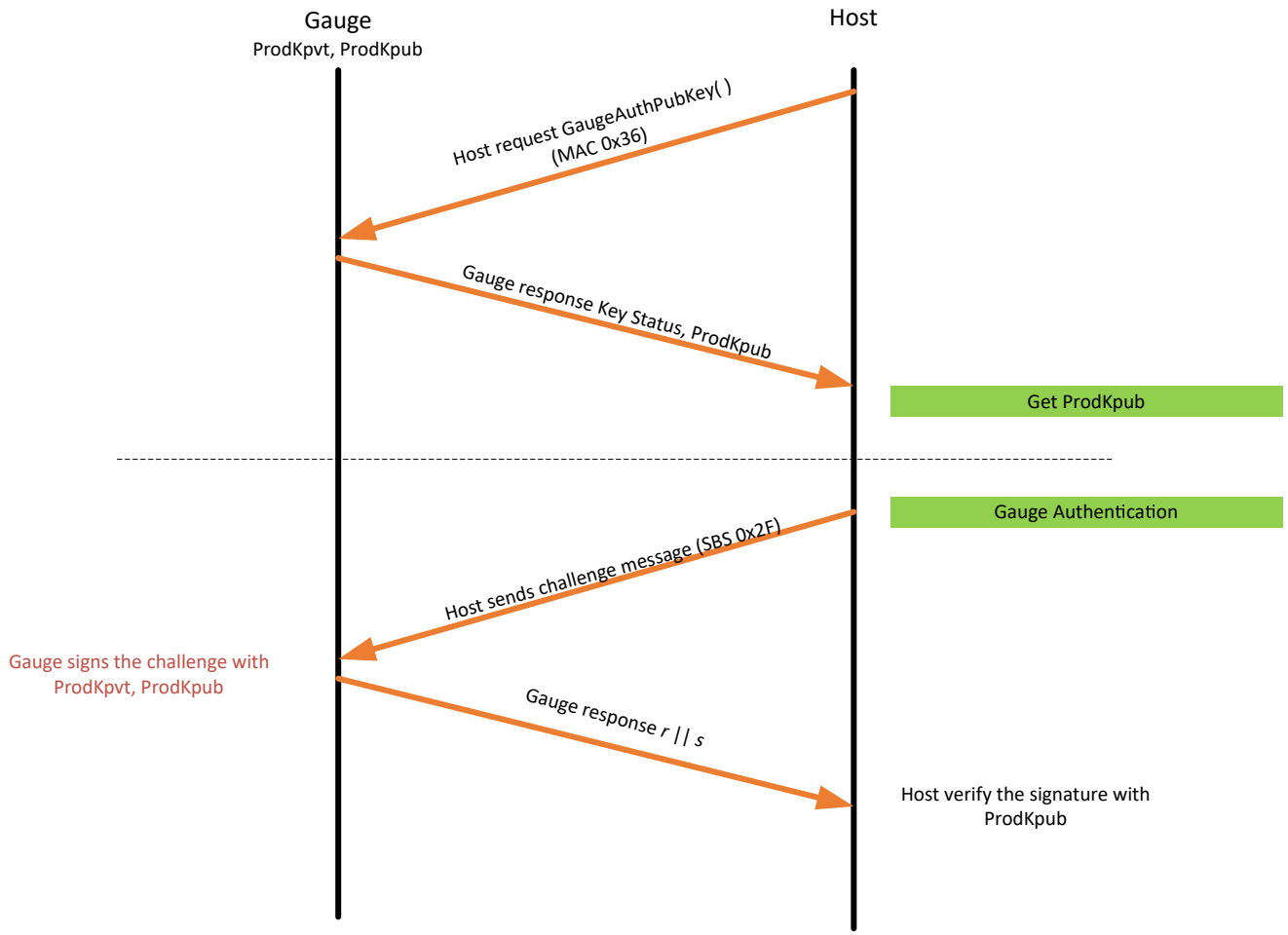
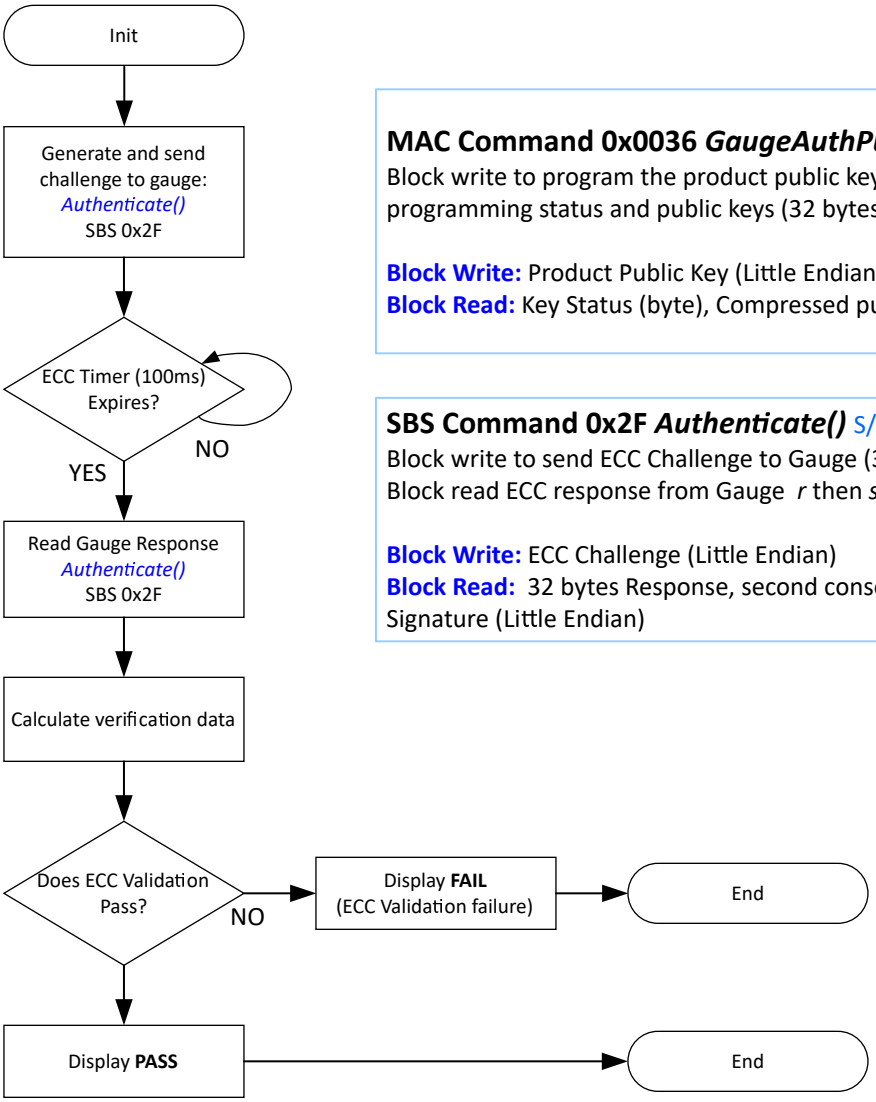


図 4-1. ゲージ認証の概要

より詳細な実装フローチャートが図 4-2 に示されています。



MAC Command 0x0036 *GaugeAuthPubKey( )* R: S/U/F, W: U/F
Block write to program the product public key and read back the programming status and public keys (32 bytes)

Block Write: Product Public Key (Little Endian)
Block Read: Key Status (byte), Compressed public key (30 bytes, LSB first)

SBS Command 0x2F *Authenticate()* S/U/F
Block write to send ECC Challenge to Gauge (30 bytes)
Block read ECC response from Gauge *r* then *s.*

Block Write: ECC Challenge (Little Endian)
Block Read: 32 bytes Response, second consecutive read 32 bytes Signature (Little Endian)

* Command availability under the following Security Modes:
S: Seal Mode
U: Unseal Mode
F: Full Access Mode

図 4-2. ゲージ認証フローチャート

5 BQ41z50 製品ファミリのホスト認証フロー

本デバイスへのアクセスをより安全なものとするため、ゲージを使用してホストを認証することもできます。これにより、ゲージはセキュリティの状態を変更し、ロックを解除して、ホストがゲージを更新できるようになります。この機能を実行するには、まずホストが許可された公開キーをプログラミングする必要があります。これには、ゲージが「フルアクセス」状態のとき、MAC サブコマンド 0x0034 に 30 バイトの圧縮された公開キーを送信し、公開キーを書き込む必要があります。ホスト認証用に現在プログラミングされている公開キーを確認するため、この同じコマンドを読み取っておく必要があります。

公開キーがプログラミングされていない場合は、古い「セキュリティキー」の UNSEAL 操作が使用されます。ただし、公開キーがプログラミングされると、古い UNSEAL コマンドは無効になります。

注

SEAL コマンドを送信する前に、プログラミング後に公開キーを読み取り、値が正しく保存されていることを確認してください。対応するプライベートキーがないと、SEALED 状態のデバイスを回復する方法はありません。

この手順は、図 5-1 図に従い、ゲージに対する SMBus ManufacturerAccess( ) (MAC) コマンドを用いて行う必要があります。

フローの概要は以下のとおりです。

1. ホストが、次のいずれかのデータブロックを用いてデータを含む MAC サブコマンド 0x003a を送信します。
   - UNSEAL アクセスアクションを要求するには、データブロック 0x14、0x04、0x72、0x36 を使用します
   - FULL_ACCESS アクセスアクションを要求するには、データブロック 0xff、0xff、0xff、0xff を使用します
2. ホストは 0x003a を使って MAC 結果を読み出し、生成された 8 バイトのチャレンジコードを受信します。
3. ホストは 8 バイトのチャレンジとコマンドを 1 つのメッセージに結合し、30 バイトの *r* と 30 バイトの *s* を生成するプライベートキーでメッセージに署名します。
   - 例えば、ゲージがチャレンジ 0x12、0x34、0x56、0x78、0x9a、0xbc、0xde、0xf0 を生成した場合、UNSEAL 操作に署名するためのフルメッセージ文字列は、0x12、0x34、0x56、0x78、0x9a、0xbc、0xde、0xf0、0x14、0x04、0x72、0x36 となります
4. ホストが、結果である *r* と *s* をゲージに書き込みます。
   - ECC_R (0x003c) サブコマンドは、30 バイトの *r* 値、またはフルの 60 バイト *r*、*s* 値を受け入れます。
   - ECC_S (0x003d) サブコマンドは、*r* 値が残量計に送信された後に書き込まれ 30 バイトの *s* 値を受け入れます。
5. ゲージに *r* と *s* の両方が設定されると、ゲージは署名を検証します。
6. 署名が有効な場合は、コマンドによって要求された作業が実行されます。

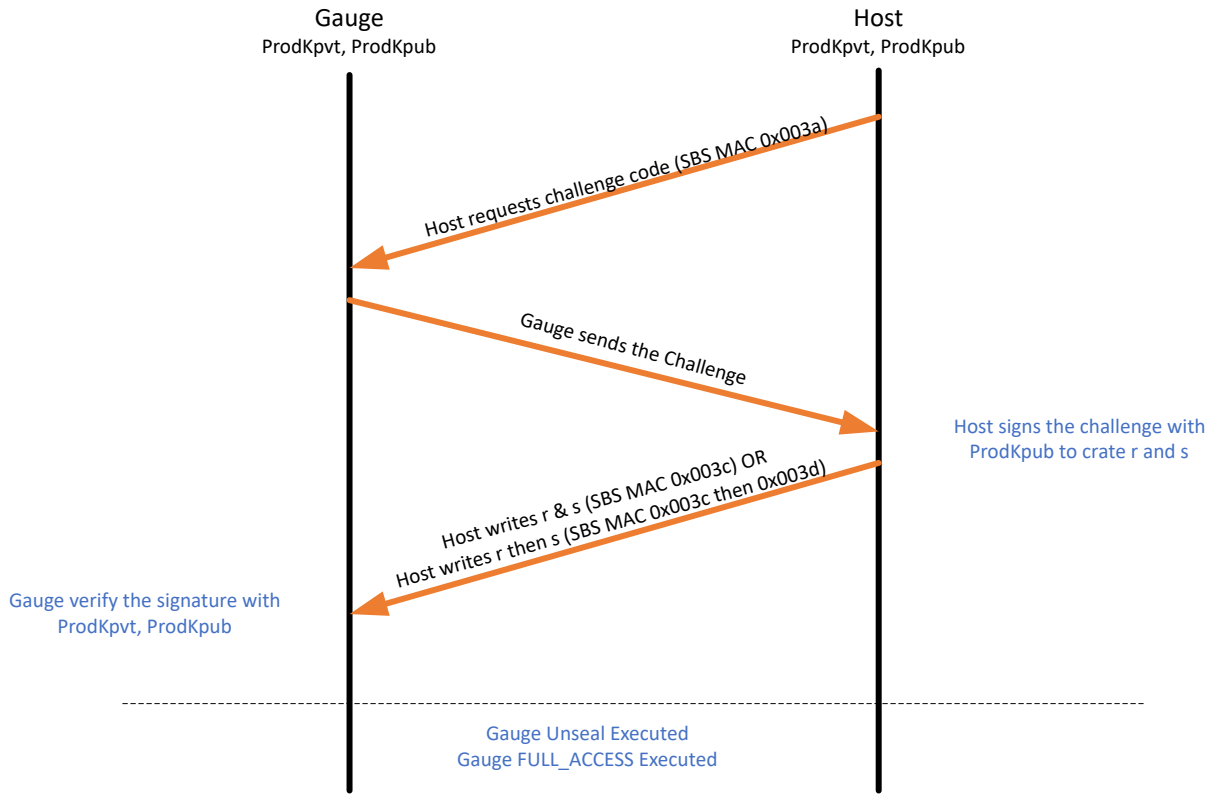
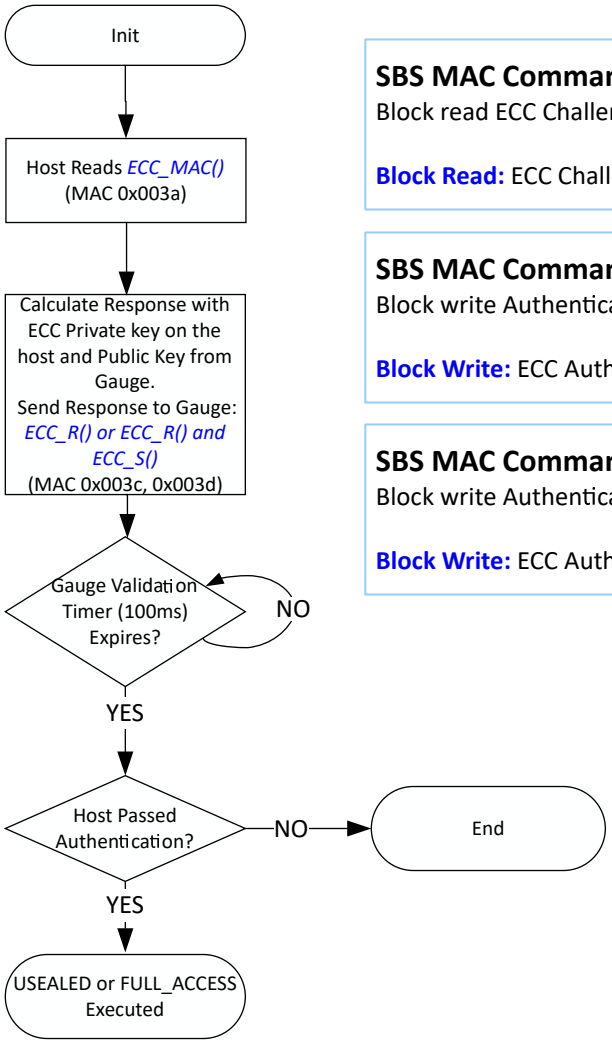


図 5-1. ホスト認証の概要

より詳細な実装フローチャートが図 5-2 に示されています。



SBS MAC Command 0x003a ECC_MAC() S/U/F
Block read ECC Challenge from Gauge (8 bytes)

Block Read: ECC Challenge (Little Endian)

SBS MAC Command 0x003c ECC_R() S/U/F
Block write Authentication *r* or *r* & *s* (30 or 60 bytes)

Block Write: ECC Authentication *r* or *r* and *s* data (Little Endian)

SBS MAC Command 0x003d ECC_S() S/U/F
Block write Authentication *s* (30 bytes)

Block Write: ECC Authentication *s* data (Little Endian)

* Command availability under the following Security Modes:
S: Seal Mode
U: Unseal Mode
F: Full Access Mode

図 5-2. ホスト認証フローチャート

6 BQSTUDIO での認証フロー

BQSTUDIO は、BQ41z50 との通信に使用できる、TI（テキサス インスツルメンツ）が提供するツールです。このドキュメントで説明する認証フローでは、*Advanced Comm SMB* タブが使用されます。図 6-1 は、ドキュメントに記載されている主なユーザーインタラクションのいくつかを説明します。

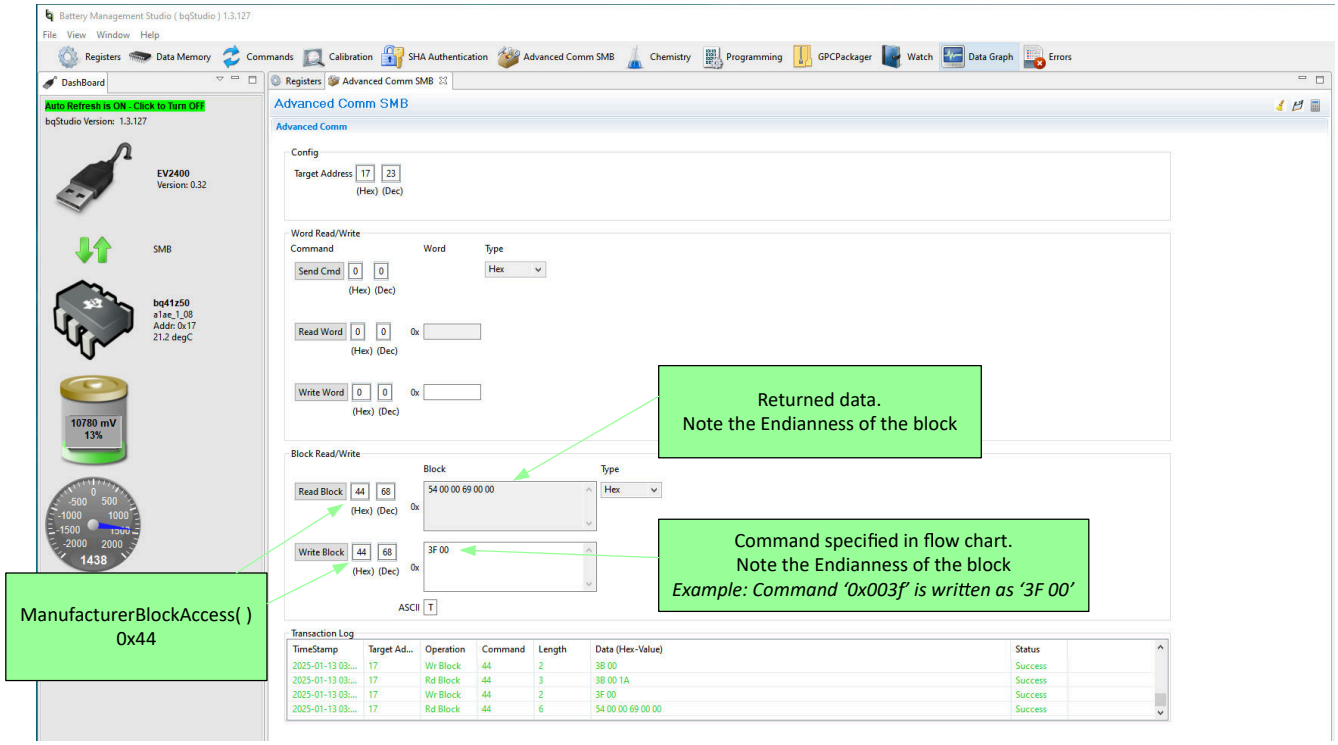


図 6-1. BQSTUDIO Advanced Comm SMB タブ

7 まとめ

要約すると、このアプリケーションノートでは、BQ41z50 デバイスの認証キーがどのように生成されるか、デバイスにどのようにキーデータがプログラミングされるか、有効化されたバッテリが bqBQ41z50 を使用してどのようにバッテリ認証に使用されるかについて説明されています。

8 参考資料

- KCDSA タスクフォースチーム、*韓国の証明書ベースデジタル署名アルゴリズム*
- 米国国立標準技術研究所、*FIPS 186-5 デジタル署名標準 (DSS)*
- 米国国立標準技術研究所、*FIPS 180-4 セキュア ハッシュ規格 (HSS)*
- 国際標準化機構、(ISO)*ISO/IEC 14888-3:2018 IT セキュリティ技術 — デジタル署名*
- テキサス・インスツルメンツ、*BQSTUDIO バッテリ マネジメント スタジオ ソフトウェア*
- テキサス インスツルメンツ、*BQ40Z50 1 ～ 4 セルの直列リチウムイオン バッテリ パック マネージャ | バッテリ残量計*
- テキサス インスツルメンツ、*BQ41Z50 2 ～ 4 個の直列リチウムイオン、高度に統合されたバッテリ残量計とプロテクタ*

表 8-1. アルゴリズムサマリーとメモ

アルゴリズム	注
ECDH	#peer_kpub は 32 バイトです #self_kpri は 32 バイトです #secret は 30 バイトです `secret = ecdh(peer_kpub, self_Kpvt)`

PBKDF2HMAC	#Openssi からの Python コードのサンプル
	#Salt は、ゲージから読み取られる乱数です
	#長さは、AES-128 の場合は 16 バイトです
	#イテレーションは 128 です
	#キーは AES-128 キーです
	
	`from cryptography.hazmat.primitives.kdf.pbkdf2 import PBKDF2HMAC`
	`kdf = PBKDF2HMAC(`
	`  aglorithm = hashes.SHA256( ),`
	`  length = 16,`
	`  salt=bytes(salt),`
	`  iterations = 128,`
	`  backed=backend`
	`)`
	`aeskey = bytearray(kdf.derive(bytes(secret)))`

AES	#iv はゲージから読み取られます
	暗号化された ProdKpvt CProdKpve は 32 バイトです
	
	`from Crypto.Cipher import AES`
	`cipher = AES.new(aeskey, AES.MODE_CTR, nonce=bytes(iv[0:8]),`
	`initial_value=bytes(iv[8:16]))`
	`  CProdKpvt = cipher.encrypt(ProdKpvt)`

ECC	ECC-233

重要なお知らせと免責事項

テキサス・インスツルメンツは、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、テキサス・インスツルメンツ製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した テキサス・インスツルメンツ製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとします。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている テキサス・インスツルメンツ製品を使用するアプリケーションの開発の目的でのみ、テキサス・インスツルメンツはその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。テキサス・インスツルメンツや第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、テキサス・インスツルメンツおよびその代理人を完全に補償するものとし、テキサス・インスツルメンツは一切の責任を拒否します。

テキサス・インスツルメンツの製品は、テキサス・インスツルメンツの販売条件、または ti.com やかかる テキサス・インスツルメンツ製品の関連資料などのいずれかを通じて提供する適用可能な条項の下で提供されています。テキサス・インスツルメンツがこれらのリソースを提供することは、適用されるテキサス・インスツルメンツの保証または他の保証の放棄の拡大や変更を意味するものではありません。

お客様がいかなる追加条項または代替条項を提案した場合でも、テキサス・インスツルメンツはそれらに異議を唱え、拒否します。

郵送先住所:Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

重要なお知らせと免責事項

テキサス・インスツルメンツは、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、 テキサス・インスツルメンツ製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した テキサス・インスツルメンツ製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとします。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている テキサス・インスツルメンツ製品を使用するアプリケーションの開発の目的でのみ、 テキサス・インスツルメンツはその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。 テキサス・インスツルメンツや第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、 テキサス・インスツルメンツおよびその代理人を完全に補償するものとし、 テキサス・インスツルメンツは一切の責任を拒否します。

テキサス・インスツルメンツの製品は、 テキサス・インスツルメンツの販売条件、または ti.com やかかる テキサス・インスツルメンツ製品の関連資料などのいずれかを通じて提供する適用可能な条項の下で提供されています。 テキサス・インスツルメンツがこれらのリソースを提供することは、適用される テキサス・インスツルメンツの保証または他の保証の放棄の拡大や変更を意味するものではありません。

お客様がいかなる追加条項または代替条項を提案した場合でも、 テキサス・インスツルメンツはそれらに異議を唱え、拒否します。

郵送先住所：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265