DI-Net 暗号化

マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	21.27.4

機能説明

VPC-DI システムは、Advanced Encryption Standard 暗号ブロック連鎖（AES CBC）アルゴリズムを使用して、異なるカード間を流れるトラフィックを暗号化します。ただし、CBC アルゴリズムには 1 つデメリットがあります。認証されていない暗号化モードを使用するため、攻撃者によってどこかの時点で暗号化トラフィックが改ざんされる可能性があるからです。この問題を回避するため、より優れた保護を提供し、データの完全性を促進する認証付き暗号化アルゴリズムが使用されます。

ガロア/カウンタモード（GCM）暗号化アルゴリズムは、この脆弱性の克服に有効な認証付き暗号化モードをサポートしています。また、復号側では、GCM は追加認証データ（AAD）を使用してペイロードを認証します。

機能の仕組み

GCM 暗号化アルゴリズムは認証されるため、DI-Net トラフィック暗号化プロセスで使用されます。これは非常に安全で、特定のキー値に対して同じ初期化ベクトル（IV）が繰り返されることはありません。*param.cfg* ファイルは、暗号化アルゴリズムの設定に使用されます。

暗号ブロック連鎖（CBC）アルゴリズムと GCM アルゴリズムではどちらも、異なる内部機能を備えたブロック暗号と排他的論理和（XOR）ロジックが使用されます。

CBC暗号化プロセスは、暗号テキストと呼ばれる以前に暗号化されたブロックとプレーンテキストと呼ばれる暗号化されていないブロックの XOR 演算、および結果のブロックのブロック暗号による暗号化から成ります。ブロック暗号を使用し、結果のブロックを前の暗号テキストブロックと XOR 演算することで、暗号化されたデータまたは暗号テキストを復号すると、プレーンテキストデータが生成されます。

（注） 最初のブロックは、前のブロックに属さず、前のブロックデータの代わりに IV を使用するため、特殊なケースとして扱われます。

GCM アルゴリズムは、カウンタモードの暗号化と認証（CTR + Auth）の組み合わせです。このアルゴリズムは、ガロア体乗算とブロック暗号のカウンタモードの動作を組み合わせ、ブロック暗号からストリーム暗号への変換を支援します。各ブロックは、キーストリームの疑似ランダム値で暗号化されます。IV 値が連続的に増加するため、各ブロックは重複しない一意の値で暗号化されます。

ガロア体乗算コンポーネントは、各ブロックを Advanced Encryption Standard（AES）標準に基づく暗号化の独自の有限体と見なします。AES GCM には、ハンドシェイク認証と追加のデータ認証が組み込まれています。また、GCM 暗号化や復号プロセスはいつでも並列化でき、組み込みの認証により、ペイロードの改ざんやパドルオラクル攻撃に対する耐性があり、CBCアルゴリズムよりも優先されます。

AES-CBC-256

マスター制御機能（CF）カードでは、**openssl** を使用して暗号化されたパスワードが生成されます。CFカード単独で、起動プロセス中にすべてのカードが使用するパスワードとシークレットコードが作成されます。すべてのパスワードには、キーと IV の生成プロセス中にスロット番号が付加されます。任意のカードで他のカードのキーと IV を生成できます。ダイナミック IV テーブルを作成する場合も、同じプロセスに従います。キーの長さはそれぞれ 256 ビットで、IV の長さは 128 ビットです。

暗号化プロセス中、送信元カードでは独自のキーが使用されますが、IV はランダムに生成されます。送信元カードの対応する IV は、IP ヘッダーの送信元アドレスと宛先アドレス、および乱数を含むハッシュ関数の出力に基づいて選択された動的 IV テーブルからの IV と XOR 演算されます。この乱数は、暗号ヘッダーに含まれます。

復号プロセス中に、接続先カードで送信元カードのスロット番号を使用して、ヘッダーからの宛先アドレスと乱数に加えて、キーと送信元アドレスが選択されてから、IVが選択されます。

AES-GCM-256

暗号化アルゴリズムを **aes-gcm-256** に変更するには、暗号化アルゴリズムの追加入力として、追加認証データ（AAD）が暗号化関数に必要です。送信元と宛先の間で転送されるものであれば、キーとIVのペアが再利用されないようにすることができます。GCMセキュリティは、この機能に準拠している必要があります。万が一、キーを持つ認証済みの暗号化関数のいずれかまたはすべてのインスタンスに対して IV が繰り返されると、実装全体が偽造攻撃に対して脆弱になります。

GCM でパケットを暗号化する際、送信元カードはCBCアルゴリズムに似たキーを選択しますが、IV は、選択された IV が特定のキーに対して一意であり、これまでに使用されたことがないことを保証するメカニズムに基づいて選択されます。AAD は暗号ヘッダーに含まれ、暗号化が完了すると、ペイロードとともに送信される前に、暗号化されたデータに認証タグ「T」が追加されます。

復号プロセス中に、GCM、キー、およびIV を使用するパケットは、暗号化プロセスと同様のメカニズムを使用して選択され、認証タグは暗号化されたデータから削除されます。AAD、キー、およびIV はすべて、ペイロードの復号に使用されます。復号後に生成された認証タグが送信元から受信した認証タグと一致する場合、データの完全性が保証され、復号プロセスは成功します。

暗号化方式（iftask_aes_gcm_encrypt）

新しい暗号化方式は次のとおりです。

- 送信元カードのスロット番号を確認します。
- 保存されている値から、このスロットのキーと IV を選択します。
- 乱数を生成します。
- 送信元 IP アドレス、宛先 IP アドレス、乱数を使用して、ダイナミック IV テーブルからの選択用に *hash_index* を生成します。
- 送信元カードのIV とダイナミック IV テーブルの IV の XOR 演算を行い、次にそれを乱数と OR 演算することで、最終的な IV を生成します。これにより、最終的な IV がキーに対して一意になるため、同じキーである IV ペアが再利用されることはありません。

（注） ダイナミックIVテーブルのサイズは64で、乱数は*uint16_t*です。

- 追加の認証データとして IP フラグメントオフセット値を選択します。
- 選択または生成されたキー、IV、および AAD を使用して暗号化します。

- 生成された乱数を暗号ヘッダーに入力します。

復号方式（iftask_aes_gcm_decrypt）

新しい復号方式は次のとおりです。

- 送信元カードのスロット番号を確認します。
- 保存されている値から、送信元スロットのキーと IV を選択します。
- 暗号ヘッダーから乱数を取得します。
- 送信元 IP アドレス、宛先 IP アドレス、および乱数を使用して、ダイナミック IV テーブルから選択する *hash_index* を作成します。
- 送信元カードの IV とダイナミック IV テーブルの IV の XOR 演算を行い、乱数と OR 演算することで、最終的な IV を生成します。
- 追加の認証データとして IP フラグメントオフセット値を選択します。
- AAD を使用して、受信したペイロードを認証します。
- 選択または生成されたキー、IV、および AAD を使用して暗号化を続行します。

制限事項

この機能の既知の制限事項と制約事項は次のとおりです。

- この機能は、すべての VPC-DI システムではなく、CUPS-DI システムのみに限定されます。
- 暗号化アルゴリズムを変更するには、リロードする必要があります。アルゴリズムは、リロードする前に、*/boot1/param.cfg* ファイルを手動で変更するか、アルゴリズム変更のための新しい CLI を使用して変更できます。その後リロードを開始します。
- 暗号化アルゴリズムはカードの起動プロセスの前に設定する必要があるため、リロード前に変更を行わずに起動設定に優先アルゴリズム含めてリロードするだけでは、アルゴリズムは変更されません。
- 小さいパケットに対する **aes-gcm-256** のコンピューティング オーバーヘッドにより、認証アルゴリズムによるパフォーマンスへの影響を評価する必要があります。

暗号化アルゴリズムの設定

暗号化アルゴリズムは、カードの起動プロセス中に、起動パラメータファイルを使用して設定されます。新しい起動フラグ値 *DI_NET_ENC_ALG* は、設定時に、起動オプションとして */boot1/param.cfg* ファイルで使用できます。

このフラグは、次の CLI を使用するか、/boot1/param.cfg ファイルを手動で編集して設定できます。手動で設定する場合は、アクティブとスタンバイのすべての CF および SF カードで同じ値に設定する必要があります。デフォルトでは、「0」は CBC で、「1」は GCM です。

（注） 変更を有効にするには、暗号化アルゴリズムを変更するたびに CP をリロードする必要があります。

CUPS で暗号化アルゴリズムを設定するには、次の設定を使用します。

```
configure
   iftask di-net-encrypt-alg di_net_encrypt_alg
end
```

注：

- **di-net-encrypt-alg**：Di-LAN トラフィックの暗号化アルゴリズムを設定します。これは、暗号化アルゴリズム名を表します。

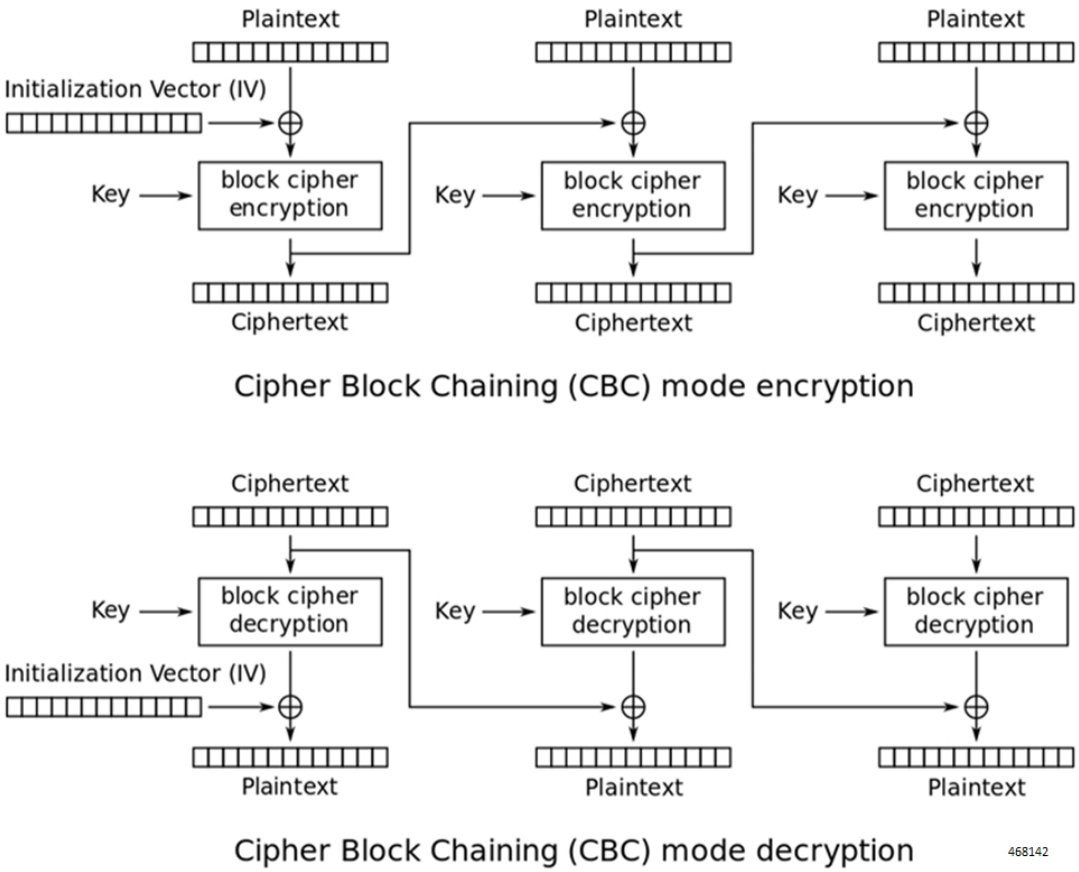
付録

暗号ブロック連鎖

暗号化中にプレーンテキストブロックと暗号テキストブロックと組み合わせる場合、機密モードでは CBC と呼ばれます。CBC には予測不能な IV が必要ですが、最初のプレーンテキストブロックと組み合わせるために、常にシークレットである必要はありません。

各プレーンテキストブロックは、前の暗号テキストブロックと XOR 演算されます。各暗号テキストブロックは、任意の時点における暗号化前のプレーンテキストブロックによって決まります。一意のテキストブロックにするためには、各メッセージ IV を最初のブロックで使用する必要があります。

図 *1:* 暗号ブロック連鎖メソッド



ガロア/カウンタモード

GCM は、暗号化のカウンタモードと新しいガロア認証モードを組み合わせたものです。主な特徴として、認証に使用されるガロア体乗算を簡単に並列化できます。

GCM を構成する 2 つの関数は、認証付き暗号化および復号と呼ばれます。認証付き暗号化関数は、機密データを暗号化し、機密データとそれ以外の非機密データの両方で認証タグを計算します。認証付き復号関数は、タグの検証を条件として、機密データを復号します。

ブロック暗号とキーが選択され承認されると、暗号化関数は次の 3 つの入力文字列を受け付けます。

- 「P」で表されるプレーンテキスト
- 追加認証付きデータ（AAD）
- 初期化ベクトル（IV）

GCM は、プレーンテキストと AAD の 2 種類のデータを、その真正性を確保することで保護します。また、AAD の透過性を維持しながら、プレーンテキストの機密性を保護します。IV は、保護する入力データに対する認証付き暗号化関数を呼び出す一意の値です。

暗号化アルゴリズムの入力文字列のビット長は、次の制限内である必要があります。

- P の長さ 239 ～ 256 以下
- A の長さ 264-1 以下
- IV の長さ 1 以上 264-1 以下

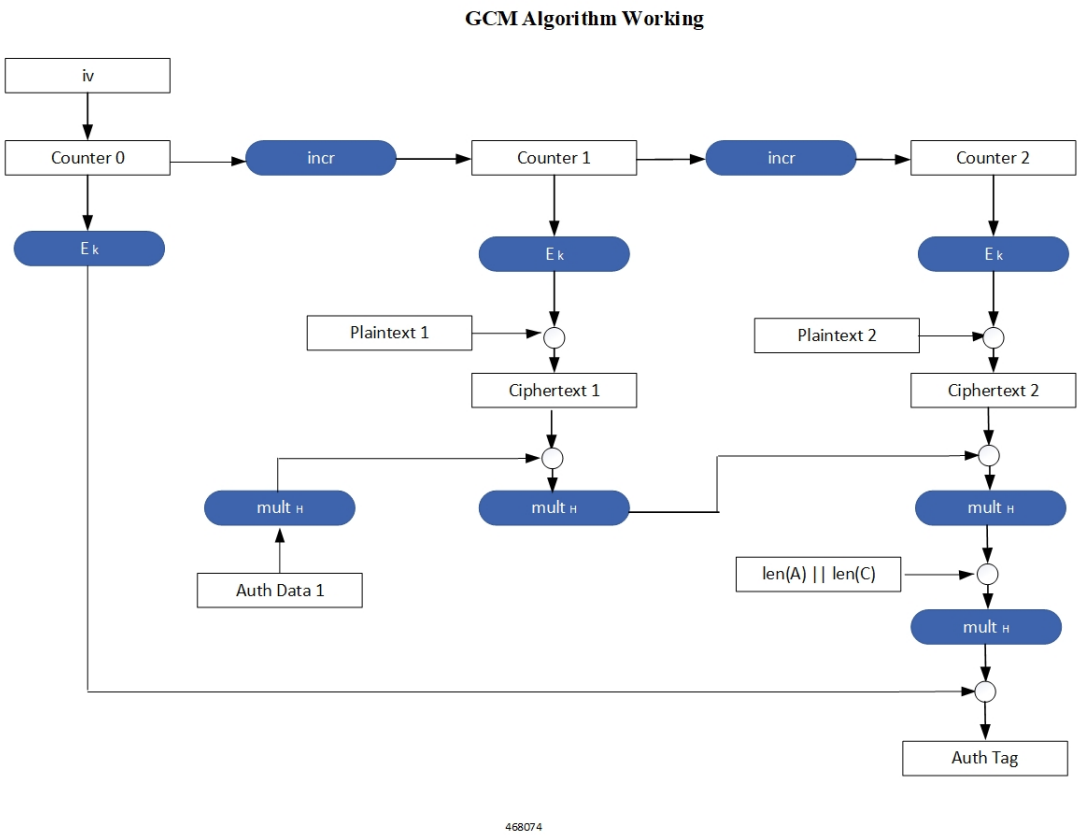
認証付き暗号化関数の入力は、IV、AAD、秘密鍵、およびプレーンテキストであり、出力は、認証タグ「T」を含むプレーンテキストと同じビット長の暗号テキストです。

ブロック暗号、鍵、および関連するタグ長を承認し選択した後、IV、追加認証付きデータ「A」、暗号テキスト「C」、および認証タグ「T」が認証付き復号関数への入力として供給されます。復号プロセスにより、次のような出力が生成されます。

- 暗号テキスト「C」に対応するプレーンテキスト「P」
- 特殊なエラーコード

（注） 出力「P」は、IV、「A」、および「C」の認証タグ「T」が成功したかどうかを示します。成功していない場合は、復号プロセスは失敗と見なされます。

図 *2:* ガロア/カウンタモードメソッド



翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。