(19)日本国特許庁(JP)

# (12)特許公報(B2)

(11)特許番号
特許第7121148号
(P7121148)

(45)発行日　令和4年8月17日(2022.8.17)

(24)登録日　令和4年8月8日(2022.8.8)

| (51)国際特許分類 | | FI | | |
|---|---|---|---|---|
| *G06F 21/60 (2013.01)* | | G06F | 21/60 | 360 |
| *H04L 9/10 (2006.01)* | | H04L | 9/10 | A |
| *H04L 9/36 (2006.01)* | | H04L | 9/36 | |
| | | G06F | 21/60 | 320 |

請求項の数　13　（全26頁）

(21)出願番号　特願2020-570718(P2020-570718)
(86)(22)出願日　令和1年6月7日(2019.6.7)
(65)公表番号　特表2021-527894(P2021-527894A)
(43)公表日　令和3年10月14日(2021.10.14)
(86)国際出願番号　PCT/US2019/036100
(87)国際公開番号　WO2019/245760
(87)国際公開日　令和1年12月26日(2019.12.26)
審査請求日　令和4年4月19日(2022.4.19)
(31)優先権主張番号　62/687,146
(32)優先日　平成30年6月19日(2018.6.19)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　16/431,548
(32)優先日　令和1年6月4日(2019.6.4)
最終頁に続く

(73)特許権者　522029730
インフィニオン テクノロジーズ エルエルシー
Ｉｎｆｉｎｅｏｎ Ｔｅｃｈｎｏｌｏｇｉｅｓ ＬＬＣ
アメリカ合衆国 カリフォルニア州 ９５１３４ サンノゼ チャンピオン コート １９８
１９８ Ｃｈａｍｐｉｏｎ Ｃｏｕｒｔ, Ｓａｎ Ｊｏｓｅ, ＣＡ ９５１３４, Ｕｎｉｔｅｄ Ｓｔａｔｅｓ ｏｆ Ａｍｅｒｉｃａ
(74)代理人　100114890
弁理士　アインゼル・フェリックス＝ラインハルト
最終頁に続く

(54)【発明の名称】　不揮発性メモリデバイス内部からの保護された通信

(57)【特許請求の範囲】
【請求項１】
メモリコントローラと、
前記メモリコントローラに結合され、サーバに通信可能に結合されたホストコンピューティングシステムによりアクセス不可能であるスタティックランダムアクセスメモリ（ＳＲＡＭ）と、
前記ホストコンピューティングシステムに結合され、処理デバイスを有する不揮発性メモリ（ＮＶＭ）デバイスと、
を有する装置であって、前記処理デバイスは、
前記ホストコンピューティングシステムを介して、セキュア通信の開始要求を有する通信パケットを前記サーバから受け取り、
セキュアプロトコルを使用し、前記ホストコンピューティングシステムを経由する通信を介して、前記サーバとセキュアハンドシェイクを実施し、
前記ホストコンピューティングシステムを介して、前記サーバからセキュアプロトコルパケット内でデータを受け取り、
前記セキュアプロトコルパケット内の暗号書き込み命令を検出することに応答して、前記セキュアプロトコルパケットを前記ＳＲＡＭの暗号バッファに格納し、
前記データを取り出すために、前記暗号バッファに格納された前記セキュアプロトコルパケットを構文解析し、
前記セキュアプロトコルパケットのヘッダから、セキュアプロトコルオペレーション識別

子とセキュアプロトコルメタデータとを取り出し、
前記セキュアプロトコルパケットの部分を、前記暗号バッファから前記ＳＲＡＭへ伝送し、
前記セキュアプロトコルパケットから取り出したセキュアプロトコルメタデータの照合を含めるために、前記セキュアプロトコルに従い、前記ＳＲＡＭから前記セキュアプロトコルパケットの前記部分を取り出して処理する、
装置。
【請求項２】
前記セキュアプロトコルは、セキュアソケット層（ＳＳＬ）プロトコルまたはトランスポート層セキュリティ（ＴＬＳ）プロトコルのうちの一方を有しており、前記セキュアハンドシェイクは、一連の暗号化オペレーションの先駆けとなる一連の順次連続したオペレーションを含む、
請求項１記載の装置。
【請求項３】
前記ＮＶＭデバイスは、フラッシュメモリデバイスであり、
前記メモリコントローラは、前記ホストコンピューティングシステムのシリアルペリフェラルインタフェース（ＳＰＩ）マスタに結合されたＳＰＩスレーブを含み、前記セキュアハンドシェイクは、前記ＳＰＩスレーブと前記ＳＰＩマスタとの間で交換されるＳＰＩパケット内のデータ伝送を介して実施され、
前記ＮＶＭデバイス内の暗号アクセラレータは、前記暗号アクセラレータにプログラミングされた暗号化ツールキットの実行を介して、暗号化オペレーションを実施するように構成されている、
請求項１記載の装置。
【請求項４】
前記処理デバイスは、さらに、
前記セキュアプロトコルパケットから取り出された少なくとも前記セキュアプロトコルメタデータを使用して、前記データを認証し、
前記ＮＶＭデバイスのＮＶＭ記憶素子に前記データを格納する、
請求項１記載の装置。
【請求項５】
前記処理デバイスは、前記ホストコンピューティングシステムによる、前記ＮＶＭ記憶素子に格納されている前記データへのアクセスを提供することができる、
請求項４記載の装置。
【請求項６】
前記セキュアプロトコルに従い前記セキュアプロトコルパケットの前記部分を処理するために、前記処理デバイスは、前記暗号アクセラレータとさらにインタラクトすることができ、これにより、
暗号スイートコードに従い前記データの１つの列を認証し、
暗号化されているならば、セッション鍵を使用し、前記データの前記列を復号して、平文データの１つの列を生成し、
平文データの前記列を前記ＮＶＭデバイスのＮＶＭ記憶素子に格納し、
前記メモリコントローラおよび前記ホストコンピューティングシステムを介して、前記ＮＶＭ記憶素子への前記データの前記列の書き込みに成功したことを、前記サーバへ折り返し報告する、
請求項３記載の装置。
【請求項７】
不揮発性メモリ（ＮＶＭ）デバイスの処理デバイスにより、前記ＮＶＭデバイスに結合されかつサーバに通信可能に結合されたホストコンピューティングシステムを介して、セキュア通信の開始要求を有する通信パケットを前記サーバから受け取るステップと、
前記処理デバイスにより、セキュアプロトコルを使用し、前記ホストコンピューティングシステムを経由する通信を介して、前記サーバとセキュアハンドシェイクを実施するス

テップと、
前記処理デバイスを用いて、前記ホストコンピューティングシステムを介して、前記サーバからセキュアプロトコルパケット内で暗号化データを受け取るステップと、
前記処理デバイスにより、前記セキュアプロトコルパケット内の暗号書き込み命令を検出することに応答して、前記セキュアプロトコルパケットを前記ＮＶＭデバイスのスタティックランダムアクセスメモリ（ＳＲＡＭ）の暗号バッファに格納するステップと、
前記処理デバイスにより、前記暗号化データを取り出すために、前記暗号バッファに格納された前記セキュアプロトコルパケットを構文解析するステップと、
前記処理デバイスにより、前記セキュアプロトコルパケットのヘッダから、セキュアプロトコルオペレーション識別子とセキュアプロトコルメタデータとを取り出すステップと、
前記処理デバイスにより、前記セキュアプロトコルパケットの部分を、前記暗号バッファから前記ＳＲＡＭへ伝送するステップと、
前記処理デバイスにより、前記セキュアプロトコルメタデータの照合を含めるために、前記セキュアプロトコルに従い、前記ＳＲＡＭから前記セキュアプロトコルパケットの前記部分を取り出して処理するステップと
を有する方法。
【請求項８】
前記セキュアハンドシェイクを実施するステップは、シリアルペリフェラルインタフェース（ＳＰＩ）パケット内のセキュアプロトコルデータを前記ホストコンピューティングシステムと交換するステップを有する、
請求項７記載の方法。
【請求項９】
前記セキュアプロトコルは、セキュアソケット層（ＳＳＬ）プロトコルまたはトランスポート層セキュリティ（ＴＬＳ）プロトコルのうちの一方を有しており、前記セキュアハンドシェイクは、一連の暗号化オペレーションの先駆けとなる一連の順次連続したオペレーションを含む、
請求項７記載の方法。
【請求項１０】
前記サーバとセキュアハンドシェイクを実施するステップは、一対のセッション鍵のうちの１つのセッション鍵の生成を有し、前記１つのセッション鍵は、前記ホストコンピューティングシステムにアクセス不可能であり、
前記方法は、
前記処理デバイスにより、前記セッション鍵を使用し、前記暗号化データを復号して、平文データを生成するステップと、
前記処理デバイスにより、前記平文データを前記ＮＶＭデバイスのＮＶＭ記憶素子に格納するステップと、
をさらに有する、
請求項７記載の方法。
【請求項１１】
前記方法は、前記ホストコンピューティングシステムによる、前記ＮＶＭ記憶素子に格納されている前記平文データへのアクセスを提供するステップをさらに有する、
請求項１０記載の方法。
【請求項１２】
前記方法は、前記セキュアプロトコルパケットから取り出された少なくとも前記セキュアプロトコルメタデータを使用して、前記暗号化データを認証するステップをさらに有する、
請求項７記載の方法。
【請求項１３】
前記処理するステップは、
前記ＮＶＭデバイスの暗号アクセラレータにより、暗号スイートコードに従い前記セキ

ュアプロトコルパケットの前記暗号化データの１つの列を認証するステップと、
　前記暗号アクセラレータにより、前記セッション鍵を使用し、前記暗号化データの前記列を復号して、平文データの１つの列を生成するステップと、
　平文データの前記列を前記ＮＶＭ記憶素子に格納するステップと、
　前記ホストコンピューティングシステムを介して、前記ＮＶＭ記憶素子への前記暗号化データの前記列の書き込みに成功したことを、前記サーバへ折り返し報告するステップと、
をさらに有する、
請求項１０記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
　関連出願
　本願は、２０１８年６月１９日出願の米国特許仮出願第６２／６８７１４６号の優先権の利益を主張する２０１９年６月４日出願の米国特許非仮出願第１６／４３１５４８号の国際出願であり、上記米国特許仮出願は、ここでの参照をもってその開示内容全体が本明細書に取り込まれたものとする。
【０００２】
　技術分野
　本開示はメモリデバイスの分野に関し、具体的には、不揮発性メモリデバイス内部からの通信を保護することに関する。
【背景技術】
【０００３】
　モノのインターネット（ＩｏＴ）デバイスは、ホストコンピューティングシステム、または外部不揮発性メモリ（ＮＶＭ）デバイスに結合されたデバイスを含み、ここで外部不揮発性メモリ（ＮＶＭ）デバイスとは例えばフラッシュメモリデバイスなどであるが、強誘電体ＲＡＭ（ＦＲＡＭ）、磁気抵抗ＲＡＭ（ＭＲＡＭ）およびこれらと同様のものなど、他のデバイスを使用してもよい。ホストコンピューティングシステム上で実行されるアプリケーションは、アプリケーションまたはブートコードをＮＶＭデバイスにダウンロードし、コードのソースを認証する能力、またはＮＶＭに書き込まれている間にコードが変更されなかったことを保証する能力がなければ、このことによってセキュリティリスクがもたらされる。
【０００４】
　いくつかの実施形態において、コードの認証および安全な出所の保証が、ホストコンピューティングシステムによるセキュアプロトコルトランザクションを介して実施される。しかしながら、多くの小型化されたＩｏＴデバイスのメモリコントローラユニットは、鍵を保護することができず、したがってセキュアプロトコルトランザクションに不正にアクセスされる。しかも、ホストコンピューティングシステムから外部ＮＶＭデバイスに書き込まれる平文データは保護されておらず、これによって攻撃者によるそのデータの読み出しおよび／または変更が可能となってしまう。その代わりに暗号化データがＮＶＭデバイスのＮＶＭに書き込まれるとしても、ホストコンピューティングシステムにおいてデータを復号しなければならず、したがってこれによりＩｏＴデバイスを再び攻撃する可能性が開かれてしまい、例えばホストコンピューティングシステムとＮＶＭデバイスの双方において、攻撃面が拡大される。
【０００５】
　添付の図面の各図には、限定ではなく例示として本開示が示されている。
【図面の簡単な説明】
【０００６】
【図１Ａ】さまざまな実施形態による、外部不揮発性メモリ（ＮＶＭ）デバイスを含むモノのインターネット（ＩｏＴ）ノードのシステムのブロック図である。
【図１Ｂ】１つの実施形態による、付加的なサブコンポーネントを示す図１Ａのシステム

のブロック図である。
【図２Ａ】１つの実施形態による、リモートコネクションとＩｏＴノード内部のシリアルペリフェラルインタフェース（ＳＰＩ）バスとを示す、図１Ａのシステムのブロック図である。
【図２Ｂ】１つの実施形態による、図２Ａに示したハードウェアに対応する単純化されたデータフロー図である。
【図３】１つの実施形態による、セキュア通信セッションを開始し、サーバとセキュアＮＶＭデバイスとの間で暗号化データを交換する方法のフローチャートである。
【図４】図４Ａは、１つの実施形態による、図１Ａ～図１Ｂのシステムの主要コンポーネントのソフトウェアおよびファームウェアのブロック図であり、図４Ｂは、１つの実施形態による、サーバとＮＶＭデバイスとの間のセキュアプロトコル通信のための方法を示す、図１Ａ～図１Ｂの上述の主要システムコンポーネントのハードウェアのブロック図である。
【図５】１つの実施形態による、サーバにより開始されたＮＶＭデバイスにおける保護された書き込み命令の実行のために、ＩｏＴデバイスのホストコンピューティングシステムとＮＶＭデバイスとにより実施されるステップを示すブロック図である。
【図６】１つの実施形態による、サーバとセキュアＮＶＭデバイスとの間でセキュア通信を確立する目的でセキュアプロトコル通信セッションを確立する方法のフローチャートである。
【図６Ａ】１つの実施形態による、例えば無線を介したファームウェアのアップデート（ｆｉｒｍｗａｒｅ ｏｖｅｒ ｔｈｅ ａｉｒ ｕｐｄａｔｅ（ＦＯＴＡ））のための、サーバとＮＶＭデバイスとの間のセキュアプロトコルハンドシェイクのフローチャートである。
【図６Ｂ】１つの実施形態による、例えばＦＯＴＡのための、サーバとＮＶＭデバイスとの間のセキュアデータ伝送のフローチャートである。
【図７】本明細書で述べる方法論のうちのいずれか１つまたは複数を機械に実施させるための命令セットを内部で実行させることのできるコンピューティングシステムの例示的な形態として、１つの機械の図式表現を示す図である。
【発明を実施するための形態】
【０００７】
ＩｏＴデバイスのサーバとＮＶＭデバイスとの間のデータ通信を保護する際の上述の欠点を解消するために（つまりはＮＶＭデバイスおよび結合されたホストコンピューティングシステムの攻撃面を閉鎖するために）、サーバとＮＶＭデバイスとの間で直接、セキュアプロトコル通信セッションを確立することができる。セキュアプロトコルを例えば、セキュアソケット層（ＳＳＬ）プロトコルまたはトランスポート層セキュリティ（ＴＬＳ）プロトコルのうちの一方とすることができる。これによりＮＶＭデバイスは、サーバを直接的に認証することができ、例えば信頼性のあるサーバからのアップデートだけしか受け入れない。この解決手段によればさらにＮＶＭデバイスは、セキュアプロトコルが相互認証を促進することから、サーバに対し自身を認証できるようになる。つまりこのことが意味するのは、ＮＶＭデバイスから受け取ったデータが途中で不正に変更されなかったことをサーバが照合できる、ということである。この解決手段によれば、セキュアプロトコル通信セッションの欠くことのできない部分として、ホストコンピューティングシステムも除外される。
【０００８】
その代わりに、あとで詳しく説明するようにさまざまな実施形態において、ホストコンピューティングシステムは、サーバからのトランスポートコントロールプロトコル（ＴＣＰ）パケットを、シリアルペリフェラルインタフェース（ＳＰＩ）パケットにパッケージングし直し、このパケットは、ＮＶＭデバイスにより認識可能であるが、ＴＣＰパケット内に当初からカプセル化されていたセキュアプロトコルパケットを依然として含む。ホストコンピューティングシステムは、ＳＰＩパケットをＮＶＭデバイスと交換して、セキュ

アプロトコルハンドシェイクおよびデータ伝送を促進することができる。実施形態において、セキュアハンドシェイクの態様を実行し、セキュアプロトコルを使用してサーバとのセキュアデータ伝送を確立する際に、（スタティックランダムアクセスメモリ（ＳＲＡＭ）の）暗号バッファおよび暗号アクセラレータ（両者ともにＮＶＭデバイスに配置されている）とインタフェース接続するために、ＮＶＭデバイスのファームウェアおよびアプリケーションプログラミングインタフェース（ＡＰＩ）をアップデートすることができる。

【０００９】

さまざまな実施形態において、ホストコンピューティングシステムは同様に、ＮＶＭデバイスからＳＰＩパケットを受け取り、ＳＰＩパケットを個々のＴＣＰパケットにカプセル化し、このパケットがサーバへ伝送される。セキュアプロトコルパケットを取り扱う際に、ホストコンピューティングシステムは、復号鍵（例えばセッション鍵の１つ）へのアクセス権がないことから、ＮＶＭデバイスまたはサーバからの暗号化データを復号することはできない。しかもホストコンピューティングシステムは、（通信の仲介として以外）意味のあるかたちでセキュアプロトコル認証に寄与しておらず、これはいまやサーバとＮＶＭデバイスとの間で直接行われる。

【００１０】

１つの実施形態において、装置は不揮発性メモリ（ＮＶＭ）デバイスを含み、これ自体はホストコンピューティングシステムに結合された処理デバイスを含む。ＮＶＭデバイスは、ＮＶＭデバイスに結合されサーバに通信可能に結合されたホストコンピューティングシステムを介して、サーバから通信パケットを受け取ることができる。この通信パケットは、セキュア通信の開始要求を伴う平文データを含む。ＮＶＭデバイスはさらに、セッション鍵（例えば一対のセッション鍵のうちの復号鍵）を生成するセキュアプロトコルを使用し、ホストコンピューティングシステムを経由した通信を介して、サーバとのセキュアハンドシェイクを実施することができる。ＮＶＭデバイスはさらに、ホストコンピューティングシステムを介してサーバから、セキュアプロトコルパケット内でデータを受け取ることができる。ＮＶＭデバイスはさらに、セキュアプロトコルパケットから取り出された少なくとも１つのセキュアプロトコルメタデータを使用して、データを認証することができる。ＮＶＭデバイスは、（暗号化されているならば）セッション鍵を使用し、データを復号して平文データを生成し、この平文データをＮＶＭデバイスのＮＶＭ記憶素子に格納することもできる。

【００１１】

実施形態において、ホストコンピューティングシステムは、（暗号化されているならば）データを復号することはできず、例えばその理由は、セキュアプロトコルセッションがサーバとＮＶＭデバイスとの間で直接開始されたため、ホストコンピューティングシステムは適切なセッション鍵（例えば復号鍵）を有していないからである。ホストコンピューティングシステムはデータを認証することもできず、その理由は、ホストコンピューティングシステムは、セキュアプロトコルパケットのセキュアプロトコルメタデータへのアクセス権を有しておらず、ＮＶＭデバイスとサーバとの間で認証のために使用されている暗号スイートを知らないからである。

【００１２】

別の実施形態において、システムはＮＶＭデバイスを含み、このＮＶＭデバイスは、ホストコンピューティングシステムを介してサーバとの接続を試み、ホストコンピューティングシステムは、サーバおよびＮＶＭデバイスとネットワーク通信を行う。実施形態において、ホストコンピューティングシステムは、サーバからＮＶＭデバイスへ、このＮＶＭデバイスとのセキュア通信の開始要求を有する平文データを含む通信パケットを伝送することができる。ホストコンピューティングシステムはさらに、ＮＶＭデバイスとサーバとの間のセキュアハンドシェイクの実施を促進することができ、このセキュアハンドシェイクは、セキュアプロトコルを使用してセキュア通信を開始し、このセキュア通信においてＮＶＭデバイスは、ホストコンピューティングシステムにはアクセス不可能な第１のセッション鍵を生成する。ホストコンピューティングシステムはさらにサーバから、セキュア

プロトコルパケットを含む伝送制御プロトコル（ＴＣＰ）パケットを受け取ることができる。セキュアプロトコルパケットは、サーバが第２のセッション鍵（例えばサーバにより生成された一対のセッション鍵のうちの暗号化鍵）を用いて暗号化した暗号化データを含む。ホストコンピューティングシステムはさらに、ＴＣＰパケットのＴＣＰヘッダを取り除いて、セキュアプロトコルパケットを露出させることができる。ホストコンピューティングシステムはさらに、シリアルペリフェラルインタフェース（ＳＰＩ）暗号書き込み命令とセキュアプロトコルオペレーション識別子とを、セキュアプロトコルパケットに追加することによって、ＳＰＩパケットを生成することができる。次いでホストコンピューティングシステムは、ＳＰＩパケットをＮＶＭデバイスへ伝送することができる。

【００１３】

このようにして、（サーバなどの）ソースとＮＶＭデバイスとの間でセキュア通信セッションを確立し使用することによって、非セキュアホストコンピューティングシステムを上回る多数の利点を提供することができる。それらの利点には、ダウンロードまたはアップロードのソースを認証するケイパビリティ、ダウンロードまたはアップロードされたデータの機密性を保証するケイパビリティ、さらにはダウンロードおよびアップロードされたデータの完全性および真正を保証するケイパビリティが含まれる。

【００１４】

実施形態において、（サーバなどの）ソースとＮＶＭデバイスとの間でセキュア通信セッションを確立し使用することによって、たとえセキュアホストコンピューティングシステムを使用したとしてもそれを上回る多数の利点を提供することができる。それらの利点には、攻撃の可能な層としてホストコンピューティングシステムを取り除くことで、セキュアコネクションの攻撃面が小さくなる、ということが含まれる。より具体的には、暗号文データとしてＮＶＭデバイスへ伝達される平文データは、通信経路沿いのいかなるポイントでも復号されず、あとでさらに詳しく説明するように、ＮＶＭデバイスのセキュア暗号バッファにいったん到達してそこにバッファリングされたときだけしか復号されない。しかも、ホストコンピューティングシステムの非セキュアユーザアプリケーションと共存するセキュアソリューションを実装することは、重要なことである。サーバとＮＶＭメモリとの間の直接的なセキュアプロトコルコネクションによって、かかる非セキュアユーザアプリケーションのためにセキュアソリューションを実装する要求が単純化される（場合によっては省かれる）。

【００１５】

図１Ａは、さまざまな実施形態による、外部不揮発性メモリ（ＮＶＭ）デバイスを含むモノのインターネット（ＩｏＴ）ノード１０１のシステム１００のブロック図である。ＩｏＴノード１０１は、いくつかの文脈ではエッジデバイスとも呼ばれる。システム１００は、ネットワーク１１５と（またはネットワーク１１５を介して）接続されており、このネットワークはクラウドとも呼ばれ、一般にはインターネットを介した１つまたは複数のバックボーンコネクションであると解することができる。システム１００はさらにサーバ１０５を含むことができ、ファームウェアアップデートを受け取る目的で、センサデータまたは他のインテリジェンスを提供する目的で、またはこれらと同様の目的で、このサーバ１０５にネットワーク１１５を介してＩｏＴノード１０１を接続することができる。サーバ１０５は一般に、伝送制御プロトコル（ＴＣＰ）／インターネットプロトコル（ＩＰ）を介して、例えばＴＣＰ／ＩＰを介して、通信すると解される。

【００１６】

さまざまな実施形態において、ＩｏＴノード１０１をマルチチップモジュールまたは半導体パッケージとして例示することができ、これは不揮発性メモリ（ＮＶＭ）デバイス１１０に結合されたホストコンピューティングシステム１０２を含む。ＮＶＭデバイス１１０を、フラッシュデバイス、ソリッドステート記憶デバイス、強誘電体ＲＡＭ（ＦＲＡＭ）、磁気抵抗ＲＡＭ（ＭＲＡＭ）、または他の不揮発性メモリデバイスとすることができる。

【００１７】

実施形態において、ホストコンピューティングシステム１０２は、他のコンポーネントもある中で特に、プロセッサ１０４、メモリコントローラユニット１０６（例えばマスタＭＣＵ）およびブリッジドライバ１０８を含む。あとでさらに詳しく説明するように、ホストコンピューティングシステム１０２は、例えばシリアルペリフェラルインタフェース（ＳＰＩ）バス、インターインテグレーテッドサーキット（Ｉ２Ｃ）バス、または他のタイプのバス伝送プロトコルなどのようなバス１１７を介して、ＮＶＭデバイス１１０に結合することができる。あとでさらに詳しく説明するように、サーバ１０５とＮＶＭデバイス１１０との間の通信を促進するために、伝送制御パケット（ＴＣＰ）をＳＰＩパケットに変換し、またこの逆を行うように、ブリッジドライバ１０８を整合することができる。

【００１８】

実施形態において、ＮＶＭデバイス１１０は、通信インタフェース１３０、マイクロコントローラ１１８（例えばＮＶＭデバイス１１０の処理デバイス）、ＮＶＭ記憶素子１２０（これをＮＶＭ記憶セルの記憶アレイとして構成することができる）、スレーブメモリコントローラ（ＳＭＣ）１２２、スタティックランダムアクセスメモリ（ＳＲＡＭ）１２６、および暗号（「クリプト」）アクセラレータ１４０を含む。通信インタフェース１３０は、読み込み／書き込みポート１３６を含むことができる。ＳＭＣ１２２は、ＳＭＣバッファ１２４と、例えばＳＰＩベースの命令を復号するＳＰＩ命令デコーダ１２５とを含むことができる。さらにＳＲＡＭ１２６は暗号バッファ１２８を含むことができ、このバッファ１２８は、暗号化オペレーションを含むＳＰＩパケットをバッファリングすることができる。実施形態において、ＳＭＣ１２２は、ホストコンピューティングシステム１０２のＭＣＵ１０６から、読み込みオペレーションおよび書き込みオペレーションを受け取り、ＳＲＡＭ１２６およびＮＶＭ１２０を参照して、読み込み命令および書き込み命令の完了を指示する。

【００１９】

図１Ｂは、１つの実施形態による、付加的なサブコンポーネントを示す図１Ａのシステム１００のブロック図である。付加的な実施形態において、ホストコンピューティングシステム１０２は、例えばＭＣＵ１０６の一部とすることができるＳＰＩマスタ１５２を含む。しかもＮＶＭデバイス１１０のＳＭＣ１２２はさらに、ＳＰＩマスタ１５２とのやりとりにおいて、ＳＰＩパケットを用いバス１１７を介して通信するためのＳＰＩスレーブ１５４を含む。表１には、ＳＰＩスレーブ１５４との通信のためにＳＰＩマスタ１５２により使用可能な例示的なアプリケーションプログラミングインタフェース（ＡＰＩ）のセットが示されており、ここで「ＳＰＩ」は、シリアルペリフェラルインタフェースのことを表し、「ＴＣＰ」は、例えばＴＣＰ／ＩＰのＴＣＰを参照する伝送制御プロトコルのことを表す。

【００２０】

【表１】

| API | 機能 |
|---|---|
| spi_write( ) | SMCバッファ124への書き込み |
| spi_read( ) | SMCバッファ124からの読み出し |
| tcp_to_spi( ) | 単一のTCPパケットをN個のSPIパケットに分割 |
| spi_to_tcp( ) | N個のSPIパケットを単一のTCPパケットに統合 |

表1

【００２１】

実施形態において、マイクロコントローラ１１８は、アドバンストハイパフォーマンスバスライト（Ａｄｖａｎｃｅｄ　Ｈｉｇｈ－Ｐｅｒｆｏｒｍａｎｃｅ　Ｂｕｓ　（ＡＨＢ）－Ｌｉｔｅ）といったシステムオンチップ（ＳｏＣ）バスアーキテクチャを介して、通信

を行う。したがって１つの実施形態において、マイクロコントローラ１１８はＡＨＢ－Ｌｉｔｅマスタ１４８を含む。ＡＨＢ－Ｌｉｔｅプロトコルは、アドバンストマイクロコントローラバスアーキテクチャ（Ａｄｖａｎｃｅｄ Ｍｉｃｒｏｃｏｎｔｒｏｌｌｅｒ Ｂｕｓ Ａｒｃｈｉｔｅｃｔｕｒｅ（ＡＭＢＡ））のオープン標準であり、これはＳｏＣ設計の機能ブロックの接続および管理のために、オンチップ相互接続仕様を提供する。本明細書ではＡＨＢ－Ｌｉｔｅを引き合いに出すけれども、他のマイクロコントローラバスアーキテクチャも考えられる。

【００２２】

相応の実施形態において、ＳＭＣ１２２および暗号アクセラレータ１４０は各々、ＡＨＢ－Ｌｉｔｅスレーブ１５６Ａおよび１５６Ｂをそれぞれ含み、ＡＨＢ－Ｌｉｔｅマスタ１４８がこれらと通信可能である。マイクロコントローラ１１８は、暗号アクセラレータ１４０と連携して動作可能であり、これによりサーバ１０５とのセキュア通信セッションを開始できるようにする暗号化オペレーションを実施して、あとでさらに詳しく述べるように、データ伝送中、暗号化データをサーバ１０５と交換する。１つの実施形態において、暗号アクセラレータ１４０は、ｍｘｃｒｙｐｔｏによりプログラミングされたフィールドプログラマブルゲートアレイ（ＦＰＧＡ）デバイスであり、ｍｘｃｒｙｐｔｏは、暗号化、認証、鍵交換、セキュアソケットオペレーション、トランスポート層セキュリティオペレーション、および他のタイプの暗号化オペレーションのためのＰｙｔｈｏｎ拡張を含む拡張ツールキットである。他のタイプの暗号化ツールキットも考えられる。

【００２３】

１つの実施形態において、表２には、ＳＭＣ１２２のＡＨＢ－Ｌｉｔｅスレーブ１５６Ａと通信するために、マイクロコントローラ１１８のＡＨＢマスタ１４８により使用される例示的なＡＰＩが示されている。１つの実施形態において、表３には、暗号アクセラレータ１４０のＡＨＢ－Ｌｉｔｅスレーブ１５６Ｂと通信するために、マイクロコントローラ１１８のＡＨＢマスタ１４８により使用される例示的なＡＰＩが示されている。

【００２４】

【表２】

| API | 機能 |
|---|---|
| smc_to_sram( ) | SMCバッファからSRAMへデータを伝送 |
| sram_to_smc( ) | SRAMからSMCバッファへデータを伝送 |

表2

【００２５】

【表３】

| API | 機能 |
|---|---|
| mxcrypto_verify_signature( ) | 署名照合のためにmxcryptoのケイパビリティを使用 |
| mxcrypto_sign( ) | データ署名のためにmxcryptoのケイパビリティを使用 |
| mxcrypto_calculate_ec_point( ) | 楕円曲線点を計算するためにmxcryptoを使用 |
| mxcrypto_sha256( ) | SHA256ダイジェストを計算 |
| decrypto_and_verify_hmac( ) | データを復号、次いでそのHMAC値をチェック |

表3

【００２６】

図２Ａは、１つの実施形態によるリモートコネクションとＩｏＴノード１０１内部のＳＰＩバス１１７とを示す、図１Ａのシステム１００のブロック図である。実施形態におい

て、サーバ１０５は、クラウド例えばネットワーク１１５への一対のリモートコネクションを介して、ＩｏＴノード１０１に通信可能に結合されている。ＩｏＴノード１０１は、ＳＰＩバス１１７により結合されたホストコンピューティングシステム１０２およびＮＶＭデバイス１１０を含むことができる。

【００２７】

図２Ｂは、１つの実施形態による、図２Ａに示したハードウェアに対応する単純化されたデータフロー図２００である。実施形態において、データフロー図２００は、サーバ１０５に格納された平文データと共にスタートするデータフローを示している。サーバ１０５は、ＴＬＳ（または他のセキュアインターネットプロトコルまたは関係する暗号化アルゴリズム）により平文データを暗号化して、暗号文データを生成することができる（２１０）。サーバ１０５は、ＴＬＳパケット内の暗号文データをパッケージングして、ＴＣＰ／ＩＰパケット内でＴＬＳパケットを送信することもできる（２２０）。

【００２８】

実施形態において、ＩｏＴノード１０１（例えばホストコンピューティングシステム１０２）は、ＴＣＰ／ＩＰパケットを受け取る（２２５）。次いでホストコンピューティングシステム１０２のブリッジドライバ１０８は、暗号文データを複数の暗号化データ部分に分割して、各暗号化データ部分から１つのＳＰＩパケットを生成することができる（２３０）。実施形態において、ＳＰＩバス１１７は、ＳＰＩパケットをＮＶＭデバイス１１０に渡す（２４０）。各ＳＰＩパケットは、暗号化部分を有するＴＬＳパケットを含むことができる（２４０）。次いでＮＶＭデバイス１１０は、例えば暗号アクセラレータ１４０を使用するなどして、暗号化部分を復号し、再び平文データを生成することができる（２５０）。復号と同時に、または復号と共に、ＮＶＭデバイス１１０はさらに、データが復号されたとき、ＮＶＭ１２０へプログラミングされる前に、一度に１行（または１列）ごとにデータを認証することができる。単純化されたこのデータフロー図２００の各ステップについては、例示的な実施態様と共に図４を参照しながらさらに詳しく説明する。

【００２９】

図３は、１つの実施形態による、セキュア通信セッションを開始し、サーバとセキュアＮＶＭデバイスとの間で暗号化データを交換する方法３００のフローチャートである。方法３００を処理ロジックによって実施することができ、この処理ロジックは、ハードウェア（例えば回路、専用ロジック、プログラマブルロジック、マイクロコードなど）、ソフトウェア（処理デバイス上で実行される命令など）、ファームウェア、またはこれらの組み合わせを有することができる。１つの実施形態において、方法３００は、ＮＶＭデバイス１１０のさまざまなコンポーネントにより実施される。

【００３０】

図３を参照すると、処理ロジックがホストコンピューティングシステム１０２を介してサーバ１０５とのコネクションを開始することから、方法３００をスタートさせることができる（３１０）。この試みに応答して方法３００をさらに続けることができ、ここで処理ロジックは、サーバに通信可能に結合されたホストコンピューティングシステム１０２を介して、サーバ１０５から通信パケットを受け取る（３２０）。この通信パケットは、セキュア通信の開始要求を含むことができる。方法３００をさらに続けることができ、ここで処理ロジックは、セッション鍵（例えば一対のセッション鍵のうちの復号鍵）を生成するセキュアプロトコルを使用し、ホストコンピューティングシステム１０２を経由した通信を介して、サーバ１０５とのセキュアハンドシェイクを実施する（３３０）。実施形態において、このセキュアハンドシェイクは、（ＳＭＣ１２２の）ＳＰＩスレーブ１５４と（ホストコンピューティングシステム１０２の）ＳＰＩマスタ１５２との間におけるＳＰＩパケット内でのデータ伝送を介して実施される。

【００３１】

引き続き図３を参照すると、方法３００をさらに続けることができ、ここで処理ロジックは、ホストコンピューティングシステム１０２を介してサーバ１０５からセキュアプロトコルパケット内で、暗号化データを受け取る（３４０）。ホストコンピューティングシ

ステム１０２は、この暗号化データを復号することができない。方法３００をさらに続けることができ、ここで処理ロジックは、セッション鍵を使用し、暗号化データを復号して平文データを生成する（３５０）。方法３００をさらに続けることができ、ここで処理ロジックは、平文データをＮＶＭデバイスのＮＶＭ記憶素子に格納する（３６０）。方法３００をさらに続けることができ、ここで処理ロジックは、ホストコンピューティングシステム１０２に対し、ＮＶＭ記憶素子に格納されている平文データへのアクセスを提供する（３７０）。

【００３２】

図４Ａは、１つの実施形態による、図１Ａ～図１Ｂのシステム１００の主要コンポーネントのソフトウェアおよびファームウェア（ＳＷ／ＦＷ）４００のブロック図である。ＳＷ／ＦＷ４００は例えば、アプリケーション４０５（ファームウェアアップデートプロデューサなど）、ＳＳＬスタックまたはＴＬＳスタック４１０、およびＴＣＰ／ＩＰスタック４１５を含むことができ、これらすべてをサーバ１０５上で実行することができる。ＳＷ／ＦＷ４００はさらに、ホストコンピューティングシステム１０２上で実行されるＴＣＰ／ＩＰスタック４２０およびＳＰＩドライバ４２５を含むことができる。ＳＰＩドライバ４２５を、ホストコンピューティングシステム１０２のブリッジドライバ１０８と同じものとすることができ、またはその中に統合することができる。ＳＷ／ＦＷ４００はさらに、ＮＶＭデバイス１１０上で実行されるＳＭＣファームウェア４３０、ＳＳＬ／ＴＬＳスタック４３５およびアプリケーション４４０（例えばファームウェアアップデートコンシューマ）を含むことができる。

【００３３】

図４Ｂは、図１Ａ～図１Ｂの上述の主要システムコンポーネントのハードウェアのブロック図であり、１つの実施形態による、ホストコンピューティングシステム１０２を媒介として使用して、サーバ１０５とＮＶＭデバイス１１０との間のセキュアプロトコル通信を行うための方法４５０を示している。１つの実施形態において、ホストコンピューティングシステムは、Ｚｙｎｑ－７０００　ＦＰＧＡであり、ＮＶＭデバイスはＫｉｎｔｅｘ－７　ＦＰＧＡである。

【００３４】

方法４５０を、以下のことからスタートさせることができる。すなわちサーバ１０５は、例えばＳＳＬ／ＴＬＳセッション内において、暗号化アルゴリズム４５３を使用し（例えばデータの認証も行うＧａｌｏｉｓ／Ｃｏｕｎｔｅｒ　Ｍｏｄｅ（ＧＣＭ）でＡｄｖａｎｃｅｄ　Ｅｎｃｒｙｐｔｉｏｎ　Ｓｔａｎｄａｒｄ（ＡＥＳ）を用い）、書き込み初期化ベクトル（ＩＶ）および暗号化鍵を入力として取り込んで、平文データ４５１を暗号文データ４５４（暗号化データとも呼ばれる）に暗号化する。暗号化鍵を、あとでさらに詳しく説明するように、ＮＶＭデバイス１１０とのハンドシェイクプロセス中に生成されるセッション鍵のうちの１つとすることができる。方法４５０をさらに続けることができ、ここでサーバ１０５は暗号文データ４５４を、ＴＣＰヘッダを有するＴＣＰパケット４５５内にカプセル化する。暗号文データ４５４のカプセル化は、ＴＬＳ（または他のセキュアプロトコル）ヘッダを含み、これらを合わせてセキュアプロトコルパケット４５６と呼ぶことができる。ＴＬＳヘッダは、ＴＬＳベースの（または他のセキュアプロトコルの）特定のメタデータを含むことができ、これは伝送に使用され、ＴＬＳベースの通信セッションに固有のものである。

【００３５】

方法４５０をさらに続けることができ、ここでサーバ１０５は、ＴＣＰパケット４５５をホストコンピューティングシステム１０２へ送信する（４５７）。したがってホストコンピューティングシステム１０２はサーバ１０５から、セキュアプロトコルパケット４５６を含むＴＣＰパケット４５５を受け取ることができる。既述のように、セキュアプロトコルパケット４５６は、暗号文データ４５４（例えば暗号化データ）とＴＬＳヘッダとを含むことができる。ホストコンピューティングシステム１０２はさらに、ＴＣＰパケット４５５のＴＣＰヘッダを取り除いて、セキュアプロトコルパケット４５６を露出させるこ

とができる。方法４５０をさらに続けることができ、ここでホストコンピューティングシステム１０２は、シリアルペリフェラルインタフェース（ＳＰＩ）暗号書き込み命令（ＣＭＤ　ＩＤ）と、セキュアプロトコルオペレーション識別子（例えばＴＬＳ　ＯＰ）とを、セキュアプロトコルパケットに追加することによって、ＳＰＩパケット４６１を生成し、このＳＰＩパケット４６１をＮＶＭデバイス１１０へ送信する（４６５）。暗号読み出し命令を同様に送信することができる一方、そのような場合にはこの暗号読み出し命令には暗号文データは添えられない。

【００３６】

方法４５０をさらに続けることができ、ここでＮＶＭデバイス１１０は、ＳＰＩパケット４６１をホストコンピューティングシステム１０２から受け取る。方法４５０をさらに続けることができ、ここでＮＶＭデバイスは、他にもある中で特に、あとで図５を参照しながら詳しく述べるように、入力として書き込み初期化ベクトルと複合鍵とを用い、復号アルゴリズム（例えば既述のＡＥＳ　ＧＣＭアルゴリズム）を使用して、暗号文データ４５４を復号する（４６９）。復号鍵を、サーバ１０５とＮＶＭデバイス１１０との間のセキュアハンドシェイク中に生成されたセッション鍵とすることができる。復号によって、元々はサーバ１０５により暗号化された平文データ４５１を生成することができる。

【００３７】

このようにしてホストコンピューティングシステム１０２は、ＴＣＰパケット４５５をＳＰＩパケット４６１に変換し、その際にこの変換をそれらのヘッダの特定の部分の入れ換えおよび／または除去により行い、次いでＳＰＩパケット４６１を、ＳＰＩバス１１７を介してＮＶＭデバイス１１０へ送信することができ、ＮＶＭデバイス１１０により読み出すことができる。このようにすることで、ホストコンピューティングシステム１０２は暗号文データ４５４を読み出さないが、これをＮＶＭデバイス１１０に回す。たとえ攻撃者がホストコンピューティングシステム１０２のところで暗号文データ４５４のアクセスを試みたとしても、復号鍵なしではこのデータは意味をなさないであろう。しかしながら復号鍵はホストコンピューティングシステム１０２には格納されておらず、ＮＶＭデバイス１１０のＳＲＡＭ１２６の暗号バッファ１２８に格納されているかぎりは、ホストコンピューティングシステム１０２にはアクセス不可能である。これによりホストコンピューティングシステムにおける攻撃面がなくなり、サーバ１０５とＮＶＭデバイス１１０との間のダイレクトなセキュア通信が大いに強化される。

【００３８】

図５は、１つの実施形態による、サーバ１０５により開始されたＮＶＭデバイス１１０における保護された書き込み命令の実行のために、（ＩｏＴデバイス１０１の）ホストコンピューティングシステム１０２とＮＶＭデバイス１１０とにより実施されるステップを示すブロック図である。図４Ｂを参照しながら述べたように、ホストコンピューティングシステム１０２は、命令識別子（ＣＭＤ　ＩＤ）により識別される暗号書き込み命令と、セキュアプロトコルオペレーション識別子（例えばＴＬＳ　ＯＰ）とを、セキュアプロトコルパケット４５６に追加することによって、ＳＰＩパケット４６１を生成することができる（図４Ｂ）。

【００３９】

さまざまな実施形態において、ＮＶＭデバイス１１０のＳＭＣ１２２におけるＳＰＩ命令デコーダ１２５は、ＳＰＩパケット内の暗号書き込み命令（ＣＭＤ　ＩＤ）を検出することができる。ＣＭＤ　ＩＤ（別のケースでは暗号読み出し命令となる場合もある）により識別された暗号化メモリオペレーションの検出に応答して、ＳＭＣ１２２は、ＳＰＩパケット（ＣＭＤ　ＩＤを除く）を暗号バッファ１２８にバッファリングすることができる。次いでマイクロコントローラ１１８はＳＰＩパケットを構文解析して、暗号化データ（例えば暗号文データ）を取り出し、さらにＴＬＳヘッダからセキュアプロトコルオペレーション識別子（ＴＬＳ　ＯＰ）およびセキュアプロトコル（またはＳＰＩ）メタデータを取り出すことができる。マイクロコントローラ１１８は、ＳＰＩパケットの一部分を暗号バッファからＳＲＡＭへ転送することができる。この時点で、マイクロコントローラ１１

８は、セキュアプロトコル（例えばこの実施例ではＴＬＳ）の実行を指示して、セキュア書き込みオペレーションを完了することができる。セキュアプロトコルの実行は、セキュアプロトコルメタデータの照合を含めるために、セキュアプロトコルに従い、ＳＲＡＭからＳＰＩパケットの一部分を取り出して処理することを含むことができる。

【００４０】

より具体的には、マイクロコントローラ１１８は暗号アクセラレータ１４０に対し、暗号文データを復号し、元々はサーバ１０５が暗号化したセキュアプロトコルパケット内の平文データ４５１（図４Ｂ）を生成するよう、指示することができる。セキュアプロトコルパケット内のデータは、（例えばソリッドステート記憶デバイスへの書き込みのための）プログラム消去（Ｐ／Ｅ）命令、または他の特定のタイプの書き込み命令、（例えばＮＶＭ１２０のユーザアレイ内の）ターゲットアドレス、ターゲットデータの長さ、およびターゲットデータ自体を含むことができる（すべて図示されている）。プログラム消去命令またはＰ／Ｅ命令は、ＮＶＭ記憶素子のユーザアレイ全体の消去を指示することができ（例えばＮＶＭ１２０がＥＥＰＲＯＭである場合）、あるいはＮＶＭ記憶素子のブロックを選択することができる（例えばＮＶＭ１２０がフラッシュメモリである場合）。いくつかの実施形態において、Ｐ／Ｅ命令を受け取らず、したがって消去命令は暗黙的であり、ＮＶＭ１２０に書き込まれるべきデータを収容するよう、ＮＶＭ１２０の十分な部分が消去される。次いでマイクロコントローラ１１８は、復号されたＳＰＩパケットからターゲットアドレスおよび長さの情報を用いて、ターゲットデータによりユーザアレイまたはユーザアレイの一部分（例えばターゲットアドレスのところにあるＮＶＭ記憶素子）をプログラミングすることができる。これがセキュア読み出し命令であったならば、ターゲットデータは存在せず、マイクロコントローラは、特定の長さのデータのターゲットアドレスのところでセキュア読み出しを実施することになる。

【００４１】

図６は、１つの実施形態による、サーバとセキュアＮＶＭデバイスとの間でセキュア通信を確立する目的でセキュアプロトコル通信セッションを確立する方法６００のフローチャートである。方法６００を処理ロジックによって実施することができ、この処理ロジックは、ハードウェア（例えば回路、専用ロジック、プログラマブルロジック、マイクロコードなど）、ソフトウェア（処理デバイス上で実行される命令など）、ファームウェア、またはこれらの組み合わせを含むことができる。１つの実施形態において、方法６００は、サーバ１０５と通信を行うＮＶＭデバイス１１０のさまざまなコンポーネントにより実施される。

【００４２】

図６を参照すると、処理ロジックがＮＶＭデバイス１１０とサーバ１０５との間のＴＣＰコネクションを確立することから、方法６００をスタートさせることができる（６１０）。方法６００をさらに続けることができ、ここで処理ロジックは、ＮＶＭデバイス１１０がサーバと通信する他の理由が存在する可能性があるけれども、サーバ１０５からのファームウェアアップデートをチェックする（６２０）。方法６００をさらに続けることができ、ここで処理ロジックは、ＨＴＭＬコードを用いてＳＳＬセッションまたはＴＬＳセッションを開始するために生じるような、セキュアプロトコルハンドシェイクを実施する（６３０）。セキュアハンドシェイクは、一連の暗号化オペレーションの先駆けとなる一連の順次連続したオペレーションを含むことができ、これにより１つまたは複数のセッション鍵（例えばＮＶＭデバイスのために少なくとも１つのセッション鍵、さらにサーバのために同じセッション鍵）が生成される。方法６００をさらに続けることができ、ここで処理ロジックは、サーバ１０５とのセキュアデータ伝送（または交換）を実施する（６７０）。方法６００をさらに続けることができ、ここで処理ロジックはＴＣＰコネクションを終了させる（６９５）。

【００４３】

図６Ａは、１つの実施形態による、例えば無線を介したファームウェアのアップデート（ＦＯＴＡ）のための、サーバと、クライアントデバイスとも呼ばれるＮＶＭデバイスと

の間の、セキュアプロトコルハンドシェイク６３０のフローチャートである。セキュアプロトコルハンドシェイク６３０を、複数のフェーズで実施することができ、これにはセキュアプロトコル通信セッションの初期化を実行するための種々の暗号化プロトコルトランザクションおよび／またはセキュアプロトコルトランザクションが含まれる。これらのトランザクションは、プロトコルおよびこのプロトコルの使用されているバージョンに応じて変わる可能性がある。したがって一般的なフレームワークについて述べ、通信メカニズムについて説明する。説明にあたり、表１～表３中に掲載したＡＰＩをもう一度参照し、さらにその他の可能なＡＰＩを提唱し、また、図１Ｂを参照しながら説明した一般的な通信フローを引き合いに出す。

【００４４】

さまざまな実施形態において、フェーズ１は、アップデートが必要であるというサーバ１０５からの返答に応答することで先に進み、このフェーズには"hello"メッセージをサーバ１０５へ送信することが含まれる（６３２）。このclient helloメッセージは、ＮＶＭデバイスがどの機能をサポート可能であるのかを、例えば暗号スイートのリストを、サーバに通知するためのものである。暗号スイートは、認証および暗号化／復号に使用可能な暗号プリミティブの組み合わせである。フェーズ１には、マイクロコントローラ１１８がtls_create_packet("client hello")を生成し、これがＳＭＣ１２２へ送信される、ということを盛り込むことができる。次いでＳＭＣ１２２は、spi_read(“client hello”)メッセージを生成することができ、これはホストコンピューティングシステム１０２へ送信される。次いでホストコンピューティングシステム１０２は、spi_read(“client hello”)メッセージをtcp_write(“client hello”)メッセージに変換することができ、これはサーバ１０５へ送信される。

【００４５】

さまざまな実施形態において、フェーズ２は、サーバ１０５がＮＶＭデバイス１１０に対し"hello"で応答する処理を進め、サーバ鍵交換および証明書署名照合を実施することができる（６３４）。このserver helloは、ＮＶＭデバイス１１０から受け取った可能な暗号スイートのリストの中からの１つの選択を含むことができ、この選択がクライアントに伝達される。さらに証明書署名照合のために、サーバ１０５はその証明書をtcp_write(“certificate”)メッセージ内で、ホストコンピューティングシステム１０２へ送信することができる。ホストコンピューティングシステム１０２は、tcp_to_spi()変換を実施してspi_write(“certificate”)を生成することができ、これはＳＭＣ１２２へ送信される。ＳＭＣ１２２はsmc_to_sram(“certificate”)を生成することができ、これはマイクロコントローラ１１８例えば処理デバイスへ送信される。マイクロコントローラ１１８は、smc_to_sram(“certificate”)をtls_process(“certificate”)メッセージに変換することができ、これによって暗号アクセラレータ１４０に対し、mxcrypto_verify_signature(“certificate”)がトリガされる。

【００４６】

実施形態において、（いま述べたことと）同様の一連のステップを実施して、サーバ鍵交換を実行することができ、これは交換の最後に暗号アクセラレータ１４０のアクセスがあろうがなかろうが、サーバによって開始される。サーバの鍵交換によって、ＮＶＭデバイスとサーバが１つの共通のセッション鍵セットを使用するように、例えばサーバのために少なくとも１つの暗号化鍵および復号鍵を使用し、さらにＮＶＭデバイス１１０のために暗号化鍵と復号鍵のもう１つのセットを使用するように、セッション鍵の交換を可能にすることができる。同様の一連のステップによって書き込み証明書要求を実施することができ、それらのステップにおいてサーバは、"certificate request"メッセージにおいてＮＶＭデバイスからクライアントの証明書を要求する。フェーズ２を完了するために、サーバ１０５はtcp_write(“server hello done”)メッセージを送信することができ、これはマイクロコントローラ１１８に報告されるに至るまで、送信および変換される。server hello doneメッセージによって、先に進むためにサーバがさらに多くの情報を必要としている、ということを表すことができる。

【００４７】
さまざまな実施形態において、フェーズ３は、証明書パケットの生成、クライアント鍵の交換、サーバとＮＶＭデバイスとの間で共有される楕円曲線点の処理を進め、暗号化署名（例えばmxcrypto_sign）がハンドシェイクメッセージに適用され、さらに暗号化アルゴリズム（例えばmxcrypto_sha256）がハンドシェイクメッセージに適用される（６３８）。実施形態において、マイクロコントローラ１１８によりクライアント鍵交換が開始され、これはサーバにおいて、例えばspi_read(“certificate”)からホストコンピューティングシステム１０２でのtcp_write(“certificate”)へのクライアント証明書の変換によって完了する。次いでマイクロコントローラ１１８は、暗号アクセラレータ１４０によってmxcrypto_sign(“handshake messages”)を開始して、セキュアプロトコル署名をハンドシェイクメッセージに適用することができる。
【００４８】
次いでマイクロコントローラ１１８は、証明書照合プロセスを開始することができ、これはＳＲＡＭを通ってＳＭＣへ、ホストコンピューティングシステムへ、さらにはサーバへと進む。証明書照合は、"certificate verify"メッセージを介した先行のメッセージの署名を含むことができる。マイクロコントローラ１１８はさらに、暗号仕様（"spec"）変更プロセスを開始することができ、これは同じ一連のコンポーネントを介し、ＳＰＩパケットとＴＣＰパケットとの間の同じ変換を伴って、サーバ１０５へ送信される。このクライアント"change cipher spec"メッセージは、ＮＶＭデバイスはサーバが選択した暗号スイートを使用する準備が整っている、ということをサーバに伝えるためのものである。マイクロコントローラ１１８はさらに、暗号アクセラレータ１４０によって暗号化アルゴリズム（例えばmxcrypto_sha256(“handshake messages”)）の適用を開始して、ＴＬＳメッセージをセッション鍵の暗号化鍵によって暗号化することができる。
【００４９】
さまざまな実施形態において、フェーズ４は、サーバ１０５がtcp_write(“change cipher spec”)をホストコンピューティングシステム１０２へ送信する処理を進め、このtcp_write(“change cipher spec”)は、tcp_to_spi()ＡＰＩを介してspi_write(“change cipher spec”)に変換されて、ＳＭＣ１２２へ送信される。サーバのこの"change cipher spec"メッセージによって、サーバはＮＶＭクライアントデバイスに対し、サーバは選択された暗号スイートを使用する準備が整っている、ということを確認通知することができる。ＮＶＭデバイスにおいて、ＳＭＣは、暗号仕様変更のための命令を、マイクロコントローラ１１８を介してＳＲＡＭへ送信することができる。次いでサーバ１０５は、"finished"メッセージを開始することができ、これはホストコンピューティングシステム１０２およびＳＭＣを介して、ＮＶＭデバイスのマイクロコントローラ１１８へ送信される。この"finished"メッセージまたは命令を、進行中の通信がこの先、ＮＶＭデバイスにおいて使用可能な暗号スイートからサーバが選択した暗号スイートによって保護されることになる、ということをＮＶＭデバイスに通知するためのものとすることができる。これによってセキュアプロトコルハンドシェイクが完了する。これらは、かかるセキュアプロトコルハンドシェイクに対するいくつかの考えられるステップであって、セキュアプロトコルハンドシェイクの当業者であれば自明のとおり、さらに追加のステップを実施してもよいし、またはこれよりも数を減らしたステップを実施してもよい。
【００５０】
図６Ｂは、１つの実施形態による、例えばＦＯＴＡのための、サーバ１０５とＮＶＭデバイス１１０との間のセキュアデータ伝送６７０のフローチャートである。サーバ１０５がtcp_write(“start firmware update”)メッセージをホストコンピューティングシステム１０２へ送信することから、セキュアデータ伝送６７０をスタートさせることができる（６７２）。ホストコンピューティングシステム１０２はtcp_to_spi()ＡＰＩを使用して、この命令をspi_write(“start firmware update”)メッセージに変換することができ、これはＳＭＣ１２２へ送信される。ＳＭＣ１２２は、このメッセージをマイクロコントローラ１１８へ送信することができる。次いでマイクロコントローラ１１８は、このメッセ

ージの（例えばＴＬＳヘッダ内の）ＴＬＳメタデータが有効であるか否かを判定することができる（６７４）。有効でなければ、マイクロコントローラ１１８はいかなるセキュアデータ伝送も中止することができ、メモリの読み出し命令または書き込み命令のリスニングに戻る。ＴＬＳメタデータが有効であるならば、サーバ１０５はＮＶＭデバイス１１０から確認通知を受け取り、列ごとにＮＶＭ１２０に書き込みをスタートさせることができる。

【００５１】

より具体的には、ホストコンピューティングデバイス１０２は、tcp_write(“row 1”)命令をＳＰＩ命令に変換し、これはＳＭＣ１２２によってマイクロコントローラ１１８へ送信される（６７６）。次いでマイクロコントローラ１１８は、暗号アクセラレータ１４０をトリガして、列１の（例えば構文解析されたデータの）暗号文データを復号し（６７８）、復号されたデータから生成されたハッシュベースのメッセージ認証コード（ＨＭＡＣ）を照合することができる（６８０）。ＨＭＡＣの照合は、照合の１つの実施態様であり、他の暗号スイートコードにより採用されているような他の照合も考えられる。ＨＭＡＣ（または他の暗号スイートコード）が照合されなかったならば、マイクロコントローラ１１８は、（前述のようにＴＬＳメタデータが照合されなかったことに対する応答の場合と同様に）データ伝送プロセスを中止することができる。マイクロコントローラ１１８が暗号アクセラレータ１４０から照合の確認通知を受け取ったならば、マイクロコントローラ１１８は、ファームウェアアップデートのデータの列（例えば列１）を、構文解析されたＳＰＩパケットにおけるターゲットアドレスに書き込むことができる（６８２）。マイクロコントローラ１１８はさらに、ＳＭＣ１２２およびホストコンピューティングシステム１０２を介して、データの列（例えば列１）の書き込みに成功した、ということをサーバ１０５に報告することができる（６８４）。次いでサーバ１０５は、データのさらなる列がファームウェアアップデート内に存在しているか否か、を判定することができる（６８８）。存在しているのであれば、サーバ１０５は次のセキュア書き込み命令を生成することができ、この命令は（本明細書で述べたように）再びＳＰＩパケットに変換され、これはＳＭＣ１２２へ送信され、場合によってはマイクロコントローラ１１８へ送信されて、さらなる列があるたびにブロック６７６～６８４が繰り返される。このようにして、サーバ１０５は、ＮＶＭデバイス１１０へのファームウェアアップデートの連続的な書き込みを制御する役割を続け、データの列が各々復号され、別の列が書き込まれる前に認証される。

【００５２】

もはや列が存在しなければ、サーバ１０５は、ファームウェアアップデートを終了させるステップに進むことができる（６９０）。例えばサーバ１０５は、tcp_write(“finish firmware update”)命令を送信することができ、この命令がマイクロコントローラ１１８まで進むと、この命令によりマイクロコントローラ１１８は暗号アクセラレータ１４０をトリガして、新たなイメージのダイジェストハッシュ、例えばmxcrypto_sha256(“new image”)を生成させる。このハッシュ結果をマイクロコントローラ１１８へ送信して戻すことができ、さらにはＳＭＣ１２２およびホストコンピューティングデバイス１０２を介してサーバへ、送信して戻すことができる。サーバ１０５は、ハッシュ結果をファームウェアアップデートのイメージの予め格納されたハッシュと比較することができ、これによりファームウェアアップデートのインストールに成功したことを確認することができる。その後、ＮＶＭデバイス１１０によるファームウェアアップデートを促進したホストコンピューティングデバイス１０２とのＴＣＰコネクションを、終了させることができる（図６のブロック６９５）。

【００５３】

本明細書で述べた方法は、主としてＮＶＭデバイス１１０におけるファームウェアアップデートの実施に関して説明されるけれども、サーバ１０５（または他のリモートコンピューティングデバイス）は、付加的な機能を盛り込むことのできる他の理由で、ＮＶＭデバイス１１０とセキュアに通信することもできる。例えば、セキュアハンドシェイクが実

施された後、サーバ１０５がアクセス制御命令を送信できるようにしてもよく、そのためにサーバ１０５に対し、サーバ１０５があたかもホストコンピューティングシステム１０２であるかのように権限が付与される。このアクセス制御命令は例えば、ＮＶＭ１２０の一部分をロックまたはロック解除する能力、ＮＶＭ１２０の一部分またはＮＶＭ１２０に格納された特定のプログラムまたはファームウェアに対する読み出し制御、実行制御および／または書き込み制御の種々の形態を設定する能力を含むことができる。

【００５４】

さらにサーバ１０５は、ＮＶＭデバイス１１０における診断プログラムをリモートで開始させることができ、またはＮＶＭデバイス１１０により生成された診断情報を少なくともリモートでセキュアに取り出すことができる。かかる診断データへのアクセスによってサーバ１０５は、ＮＶＭデバイス１１０が不正にアクセスされたことが、ハードウェアおよび／またはソフトウェアの機能の何らかの見地により表されているか否かを、迅速に突き止めることができるようになり、したがってサーバ１０５とＮＶＭデバイス１１０との間に確立されたセキュアネットワークの切断を保証することができる。セキュアネットワークセッションが切断されたならば、図６、図６Ａおよび図６Ｂの方法がリスタートさせられて、新たなセキュアセッションが確立される。セキュアネットワークセッションを切断せざるを得ない事態を引き起こしたハードウェアまたはソフトウェアのどのような問題であれ、セキュアデータ伝送（図６Ｂ）における機密データの交換前には解決済みである、ということをチェックするために、この方法をアップデートすることができる。

【００５５】

図７には、本明細書で述べた方法論のうちのいずれか１つまたは複数を機械に実施させるための命令セットを内部で実行させることのできるコンピューティングシステム７００の例示的な形態として、１つの機械の図式表現が示されている。代替実施態様において、ＬＡＮ、イントラネット、エクストラネットまたはインターネットを介して、この機械を他の機械と接続（例えばネットワーク化）することができる。この機械は、クライアントサーバネットワーク環境におけるサーバまたはクライアントデバイスという役割形態で、またはピアツーピア（または分散）ネットワーク環境におけるピアマシンとして、動作することができる。この機械を、ホストコンピューティングシステムまたはコンピュータ、自動車コンピューティングデバイス、コントローラエリアネットワーク（ＣＡＮ）またはローカル相互接続ネットワーク（ＬＩＮ）などの自動車ネットワークのためのサーバ、ネットワークデバイスとすることができ、あるいはこの機械がとるべきアクションを指定する命令セットを（シーケンシャルにまたは他の形態で）実行可能な任意の機械とすることができる。さらに、ただ１つの機械だけしか示されていないけれども、用語「機械」を、本明細書で述べた方法論のうちのいずれか１つまたは複数を実施するための１つの命令セット（または複数の命令セット）を、個別にまたは共働して実行する複数の機械から成る任意の集合体を含むものであるとも解されたい。変換のページおよびセクションの実施態様を、コンピューティングシステム７００において実装することができる。

【００５６】

コンピューティングシステム７００は、処理デバイス７０２、メインメモリ７０４（例えばリードオンリーメモリ（ＲＯＭ）、フラッシュメモリ、ダイナミックランダムアクセスメモリ（ＤＲＡＭ）（シンクロナスＤＲＡＭ（ＳＤＲＡＭ）またはＤＲＡＭ（ＲＤＲＡＭ）など）、スタティックメモリ７０６（例えばフラッシュメモリ、スタティックランダムアクセスメモリ（ＳＲＡＭ）など）、およびデータ記憶デバイス７１８を含み、これらはバス７３０を介して互いに通信し合う。

【００５７】

処理デバイス７０２は、マイクロプロセッサデバイス、中央処理ユニットまたはこれらに類するものなど、１つまたは複数の汎用の処理デバイスを表す。さらに詳しくは処理デバイスを、複合命令セットコンピューティング（ＣＩＳＣ）マイクロプロセッサデバイス、縮小命令セットコンピュータ（ＲＩＳＣ）マイクロプロセッサデバイス、超長命令語（ＶＬＩＷ）マイクロプロセッサデバイス、または他の命令セットを実行する処理デバイス

または命令セットの組み合わせを実行する処理デバイスとすることができる。処理デバイス７０２を、特定用途向け集積回路（ＡＳＩＣ）、フィールドプログラマブルゲートアレイ（ＦＰＧＡ）、ディジタル信号処理デバイス（ＤＳＰ）、ネットワーク処理デバイス、またはこれらに類するものなど、１つまたは複数の専用処理デバイスとすることもできる。１つの実施態様において、処理デバイス７０２は、１つまたは複数の処理デバイスコアを含むことができる。処理デバイス７０２は、本明細書で述べたオペレーションを実施するための命令７２６を実行するように構成されている。１つの実施態様において、処理デバイス７０２を、サーバ１０５、ホストコンピューティングシステム１０２、またはＮＶＭデバイス１１０の一部とすることができる。

【００５８】

別の選択肢として、コンピューティングシステム７００は、本明細書で述べたような他のコンポーネントを含むことができる。コンピューティングシステム７００はさらに、ネットワーク７２０に通信可能に結合されたネットワークインタフェースデバイス７０８を含むことができる。コンピューティングシステム７００は、ビデオディスプレイユニット７１０（例えば液晶ディスプレイ（ＬＣＤ））、英数字入力デバイス７１２（例えばキーボード）、カーソル制御デバイス７１４（例えばマウス）、信号発生デバイス７１６（例えばスピーカ）、または他の周辺デバイスを含むこともできる。さらにコンピューティングシステム７００は、グラフィック処理ユニット７２２、ビデオ処理ユニット７２８、およびオーディオ処理ユニット７３２を含むことができる。別の実施態様において、コンピューティングシステム７００はチップセット（図示せず）を含むことができ、これは処理デバイス７０２と共に動作するように設計された一群の集積回路またはチップのことを指し、処理デバイス７０２と外部デバイスとの間の通信を制御する。例えばチップセットをマザーボード上のチップセットとすることができ、マザーボードは処理デバイス７０２を、メインメモリ７０４およびグラフィックコントローラといった超高速デバイスとつなぎ、同様に処理デバイス７０２を、ＵＳＢ、ＰＣＩバスまたはＩＳＡバスといった周辺機器の低速ペリフェラルバスとつなぐ。

【００５９】

データ記憶デバイス７１８はコンピュータ可読記憶媒体７２４を含むことができ、この記憶媒体７２４に、本明細書で述べた機能の方法論のうちのいずれか１つまたは複数を具現化する命令７２６が格納される。これらの命令７２６をすべてまたは少なくとも部分的に、コンピューティングシステム７００によるこれらの実行中、命令７２６としてメインメモリ７０４内に、かつ／または処理ロジックとして処理デバイス７０２内に、常駐させることもできる。つまりこの場合、メインメモリ７０４および処理デバイス７０２も、コンピュータ可読記憶媒体を構成する。

【００６０】

図１Ａ～図１Ｂを参照して説明したように処理デバイス７０２を使用する命令７２６を格納するために、かつ／または上述のアプリケーションをコールする方法を含むソフトウェアライブラリを格納するために、コンピュータ可読記憶媒体７２４を用いることもできる。コンピュータ可読記憶媒体７２４は、例示的な実施態様において単一な媒体であるように示されているけれども、用語「コンピュータ可読記憶媒体」を、１つまたは複数の命令セットを格納する単一の媒体または複数の媒体（例えば集中型または分散型のデータベースおよび／または関連づけられたキャッシュおよびサーバ）を含むものであると解されたい。用語「コンピュータ可読記憶媒体」を、機械により実行するための命令セットを格納または符号化または搬送することができ、かつ実施態様の方法論のうちのいずれか１つまたは複数を機械に実施させる任意の媒体を含むものであるとも解されたい。したがって用語「コンピュータ可読記憶媒体」を、以下に限定されるものではないが、ソリッドステートメモリ、ならびに光学媒体および磁気媒体を含むものであると解されたい。

【００６１】

上述の記載において、数多くの細部について述べられている。しかしながら本開示の利益に与る当業者には自明のとおり、本開示の実施形態をそれらの具体的な細部がなくても

実施することができる。いくつかの事例では、説明が不明瞭になるのを避ける目的で、周知の構造およびデバイスについては、詳細図ではなくブロック図の形態で示されている。

【００６２】

本明細書で使用されるモジュールは、ハードウェア、ソフトウェアおよび／またはファームウェアの任意の組み合わせのことを指す。一例としてモジュールは、マイクロコントローラなどのハードウェアを含み、これはマイクロコントローラにより実行されるように整合されたコードを格納するために、非一時的媒体と関連づけられている。したがって１つの実施態様において、あるモジュールへの言及は、非一時的媒体上に保持されるべきコードを認識および／または実行するように特に構成されているハードウェアのことを指す。さらに別の実施態様において、あるモジュールの使用は、予め定められたオペレーションを実施するためにマイクロコントローラにより実行されるように特に整合されているコードを含む、非一時的媒体のことを指す。ここで推察できるように、さらに別の実施態様において、（この例における）モジュールという用語は、マイクロコントローラと非一時的媒体との組み合わせのことを指す場合がある。多くの場合、別個のものとして示されている各モジュールの境界は一般には可変であり、場合によってはオーバラップしている。例えば第１のモジュールおよび第２のモジュールは、場合によってはいくつかの独立したハードウェア、ソフトウェアまたはファームウェアをそのまま残しながら、ハードウェア、ソフトウェア、ファームウェアまたはこれらの組み合わせを共有することができる。１つの実施態様において、ロジックという用語の使用は、トランジスタ、レジスタといったハードウェア、またはプログラマブルロジックデバイスといった他のハードウェアを含む。

【００６３】

「するように構成されている」というフレーズの使用は、１つの実施態様において、指定または決定されたタスクを実行するために、装置、ハードウェア、ロジックまたは素子を配置すること、組み立てること、製造すること、販売に供すること、輸入すること、および／または設計することを指す。この例の場合、装置またはこの装置の素子は、それが指定されたタスクを実施するように設計、結合および／または相互接続されているならば、動作中でなくてもやはり、その指定されたタスクを実施するように「構成されている」。単に例示的な一例として、ロジックゲートは動作中、０または１を供給することができる。ただし、イネーブル信号をクロックに供給するように「構成された」ロジックゲートは、１または０を供給可能な考えられるあらゆるロジックゲートを含まない。その代わりにロジックゲートは、動作中に１または０の出力がクロックをイネーブルにすることができるよう、何らかの手法で結合されたものである。ここで再度留意されたいのは、「するように構成されている」という用語を使用するために動作は要求されないが、その代わりに、装置、ハードウェアおよび／または素子の潜在的な状態に重点が置かれ、この場合、潜在的な状態において装置、ハードウェアおよび／または素子は、装置、ハードウェアおよび／または素子が動作中であるときに、特定のタスクを実施するように設計されている、ということである。

【００６４】

さらに、「するための」、「することができる」および／または「するように動作可能である」というフレーズの使用は、１つの実施態様の場合には、何らかの装置、ロジック、ハードウェアおよび／または素子の使用を指定された手法で可能にするように設計された装置、ロジック、ハードウェアおよび／または素子のことを指す。上述のようにここで留意されたいのは、「するための」、「することができる」、または「するために動作可能である」の使用は、１つの実施態様の場合には、装置、ロジック、ハードウェアおよび／または素子の潜在的な状態のことを指し、この場合、装置、ロジック、ハードウェアおよび／または素子は動作していないけれども、装置の使用を指定された手法で可能にするように設計されている、ということである。

【００６５】

本明細書で使用される値は、数字、状態、論理状態またはバイナリ論理状態の任意の既知の表現を含む。多くの場合、論理レベル、論理値または論理的な値の使用は、「１」お

よび「０」とも呼ばれ、これは単にバイナリロジック状態を表現する。例えば１はハイ論理レベルのことを指し、０はロー論理レベルのことを指す。１つの実施態様において、トランジスタまたはフラッシュセルといった記憶セルを、単一の論理値または複数の論理値を保持可能なものとすることができる。ただしコンピュータシステムでは、これとは別の値の表現が用いられてきた。例えば１０進数の数字１０を、２進数の値１０１０および１６進数の文字Ａとして表現することもできる。したがってある１つの値は、コンピュータシステムに保持することのできる情報の任意の表現を含む。

【００６６】

詳細な説明のいくつかの部分は、コンピュータメモリ内のデータビットにおけるオペレーションのアルゴリズムおよびシンボル表現で呈示されている。アルゴリズムによるそれらの記述および表現は、データ処理の当業者により、自身の仕事の内容を他の当業者に最も有効に伝えるために用いられる手段である。ある１つのアルゴリズムは本明細書では、そして一般的に、所望の結果をもたらす首尾一貫したステップシーケンスであると考えられている。これらのステップは、物理量の物理的操作を要求するステップである。必須ではないけれどもこれらの量は通常、格納、伝送、組み合わせ、比較、およびその他の操作が可能な電気的または磁気的な信号の形態をとる。これらの信号をビット、値、要素、シンボル、キャラクタ、用語、数字、またはこれらに類するものと呼ぶのが、主として一般的な用法ゆえに、時には便利であることが判明した。

【００６７】

ただし、これらのすべておよび同様の用語は、適切な物理量と関連づけられるべきであり、それらの量に適用される便利な標号であるにすぎない、ということに留意されたい。特段の記載がないかぎり、これまでの説明から明らかなように、以下のことは自明である。すなわち説明全体を通して、「受け取る」、「調節する」、またはこれらに類するもののような用語を使った説明は、コンピューティングシステムまたは同様の電子コンピューティングデバイスのアクションおよびプロセスのことを指し、このコンピューティングシステムまたは同様の電子コンピューティングデバイスは、コンピュータシステムのレジスタおよびメモリ内の物理量（例えば電子的な量）として表現されるデータを操作し、コンピュータシステムのメモリまたはレジスタ、あるいは他のかかる情報の記憶デバイス、伝送デバイスまたは表示デバイスの内部の物理量として同様に表される他のデータへ変換する。

【００６８】

「例」または「例示的」という言葉は本明細書では、１つの例、事例または例証としての役割を果たすことを意味するために使用される。本明細書において「例」または「例示的」として述べるいずれの態様または設計も、必ずしも他の態様または設計よりも好ましいまたは有利であると捉えるべきではない。むしろ、「例」または「例示的」という言葉の使用は、コンセプトを具体的な手法で呈示することを意図している。本願において使用される用語「または」は、排他的な「または」ではなく包含的な「または」を意味することを意図している。つまり特段の記載がないかぎり、または文脈から明らかでないかぎり、「ＸはＡまたはＢを含む」は、自然な包括的順列のいずれも意味することを意図している。つまりＸはＡを含む、ＸはＢを含む、またはＸはＡとＢの両方を含む、としたならば、「ＸはＡまたはＢを含む」は、上述の事例のいずれのもとでも充足される。これに加え、本明細書および添付の特許請求の範囲において用いられている不定冠詞は、特段の記載がないかぎり、または単数形を指示していると文脈から明らかでないかぎり、一般に「１つまたは複数」を意味すると解されたい。さらに用語「ある実施形態」または「１つの実施形態」の使用、あるいは「ある実施形態」または「１つの実施形態」は全体を通して、その旨の説明がないかぎりは、同じ実施形態または実施形態を意味することを意図していない。

【００６９】

本明細書で述べた実施形態は、本明細書のオペレーションを実施するための装置にも関係するものとすることができる。この装置を、要求された目的のために特別に構築するこ

とができ、またはこの装置は、内部に格納されたファームウェアにより選択的にアクティベートまたはリコンフィギュレーションされる汎用のハードウェアを有することができる。かかるファームウェアを、非一時的なコンピュータ可読記憶媒体に格納することができ、このような記憶媒体は例えば、以下に限定されるものではないが、ＮＶＭ、リードオンリーメモリ（ＲＯＭ）、ランダムアクセスメモリ（ＲＡＭ）、ＥＰＲＯＭ、ＥＥＰＲＯＭ、フラッシュメモリ、または電子的な命令を格納するために適した任意のタイプの媒体などである。用語「コンピュータ可読記憶媒体」を、１つまたは複数の命令セットを格納する単一の媒体または複数の媒体を含むものであると解されたい。用語「コンピュータ可読媒体」を、ハードウェアにより実行するための命令セットを格納、符号化または搬送することができ、かつ本実施形態の方法論のうちのいずれか１つまたは複数をハードウェアに実施させる任意の媒体を含むものであるとも解されたい。したがって用語「コンピュータ可読記憶媒体」を、以下に限定されるものではないが、ソリッドステートメモリ、光学媒体、電磁媒体、ハードウェアにより実行するための命令セットを格納することができ、かつ本実施形態の方法論のいずれか１つまたは複数をハードウェアに実施させる任意の媒体を含むものであると解されたい。

【００７０】

本開示のいくつかの実施形態の理解を深めることができるようにする目的で、上述の記載によれば、特定のシステム、コンポーネント、方法などの実施例といった多数の具体的な細部について述べられている。ただし当業者には自明であるとおり、本開示の少なくともいくつかの実施形態を、それらの具体的な細部を伴わずに実施することができる。他の事例において、周知のコンポーネントまたは方法について詳しくは説明されておらず、あるいは本開示が不必要に不明確になるのを避ける目的で、単純なブロック図の形態で呈示されている。よって、上述の具体的な細部は、例示的なものであるにすぎない。固有の実施形態を、これらの例示的な細部から変更してもよく、それでもなお本開示の範囲内にあるものと考えることができる。

【００７１】

上述の記載は、例示的なものであり限定的なものではないことを意図している、という点を理解されたい。上述の記載を読み理解することで、当業者には他の多くの実施形態が明らかになるであろう。よって、本開示の範囲は、添付の特許請求の範囲を、かかる特許請求の範囲に権利が与えられた等価物の範囲全体と併せて参酌することで決定されるべきものである。

【００７２】

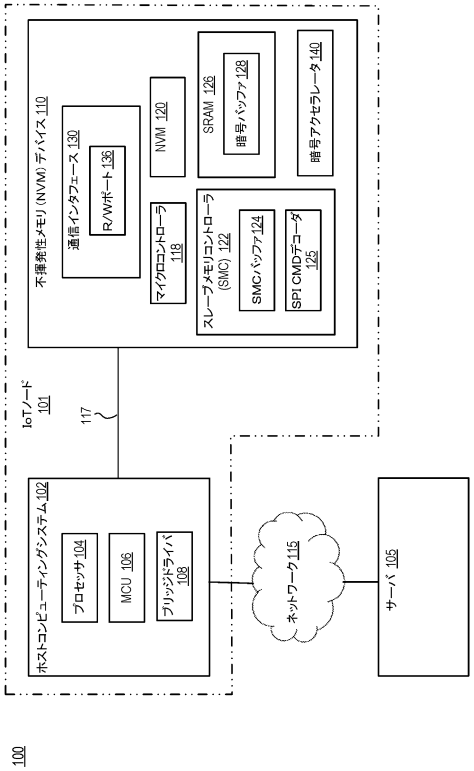
上述の記載において説明の目的で、本開示を完全に理解できるようにするために、数多くの具体的な細部について述べられている。ただし当業者には自明であるとおり、本開示をそれらの具体的な細部を伴わずに実施することができる。他の事例において、本明細書の理解を不必要に曖昧にするのを避ける目的で、周知の回路、構造および技術は、詳細にではなくブロック図で示されている。

【００７３】

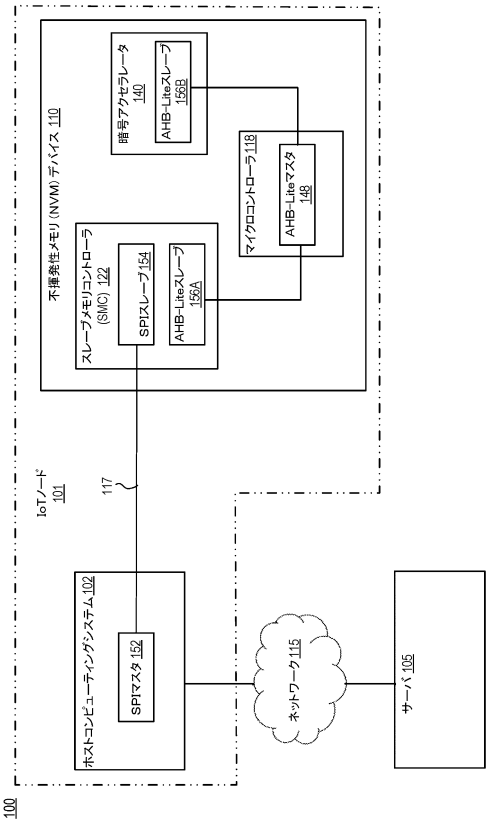
本明細書で「１つの実施形態」または「ある実施形態」について言及することは、この実施形態に関連して記載される固有の特徴、構造または特性が、本開示の少なくとも１つの実施形態に含まれている、ということを意味する。本明細書のさまざまな個所に記載された「１つの実施形態において」というフレーズは、必ずしも同じ実施形態のことを指すものではない。

【図面】

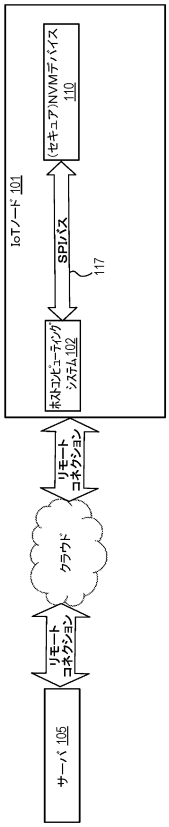
【図１Ａ】


100
IoTノード 101
不揮発性メモリ(NVM)デバイス 110
通信インタフェース 130
R/Wポート 136
NVM 120
SRAM 126
暗号バッファ 128
暗号アクセラレータ 140
マイクロコントローラ 118
スレーブメモリコントローラ (SMC) 122
SMCバッファ 124
SPI CMDデコーダ 125
117
ホストコンピューティングシステム 102
プロセッサ 104
MCU 106
プリンシドライバ 108
ネットワーク 115
サーバ 105

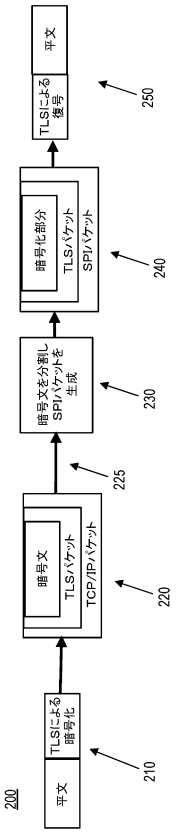
【図１Ｂ】


100
IoTノード 101
不揮発性メモリ(NVM)デバイス 110
暗号アクセラレータ 140
AHB-Liteスレーブ 156B
マイクロコントローラ 118
AHB-Liteマスタ 148
スレーブメモリコントローラ (SMC) 122
SPIスレーブ 154
AHB-Liteスレーブ 156A
117
ホストコンピューティングシステム 102
SPIマスタ 152
ネットワーク 115
サーバ 105

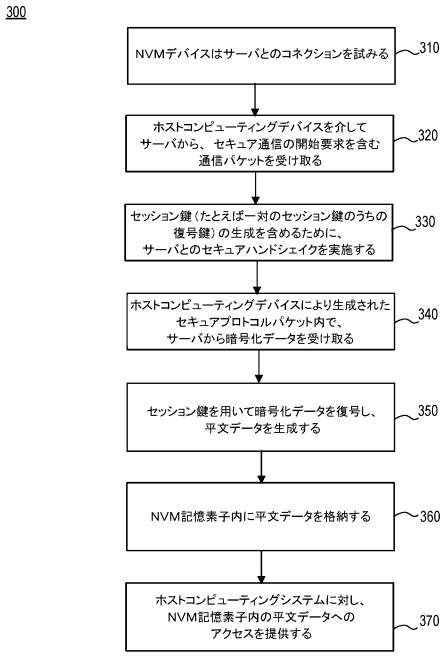
【図２Ａ】


IoTノード 101
セキュアNVMデバイス 110
SPIバス
117
ホストコンピューティングシステム 102
リモートコネクション
クラウド
リモートコネクション
サーバ 105

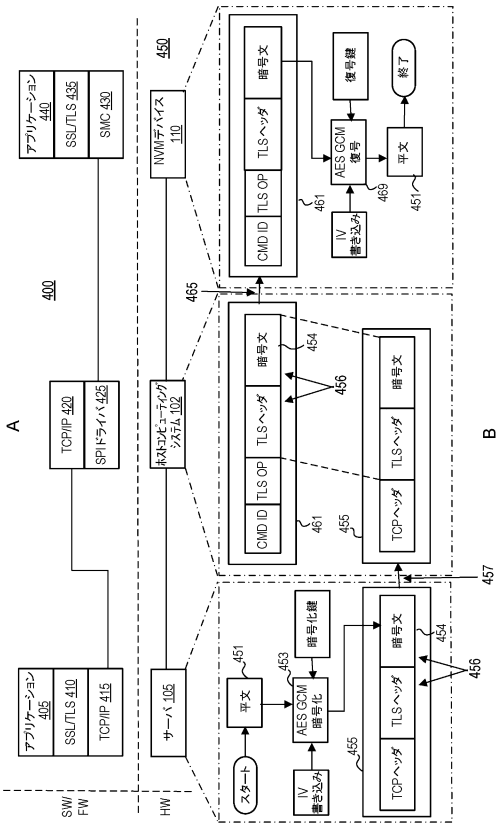
【図２Ｂ】


200
平文
TLSによる暗号化
210
暗号文
TLSパケット
TCP/IPパケット
220
225
暗号文を分割しSPIパケットを生成
230
暗号化部分
TLSパケット
SPIパケット
240
TLSによる復号
平文
250

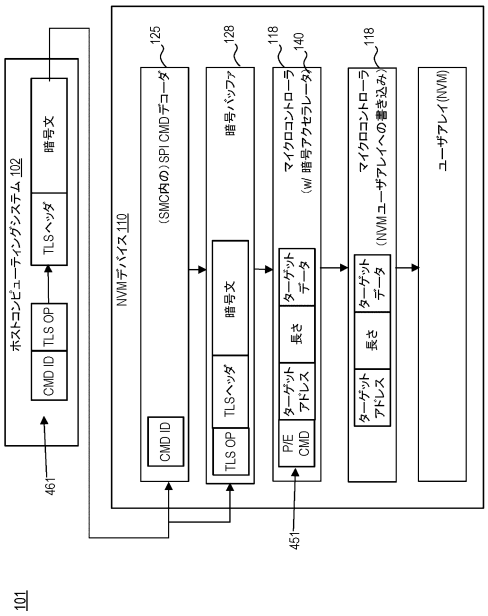
【図３】


300
NVMデバイスはサーバとのコネクションを試みる
310
ホストコンピューティングデバイスを介してサーバから、セキュア通信の開始要求を含む通信パケットを受け取る
320
セッション鍵（たとえば一対のセッション鍵のうちの復号鍵）の生成を含めるために、サーバとのセキュアハンドシェイクを実施する
330
ホストコンピューティングデバイスにより生成されたセキュアプロトコルパケット内で、サーバから暗号化データを受け取る
340
セッション鍵を用いて暗号化データを復号し、平文データを生成する
350
NVM記憶素子内に平文データを格納する
360
ホストコンピューティングシステムに対し、NVM記憶素子内の平文データへのアクセスを提供する
370

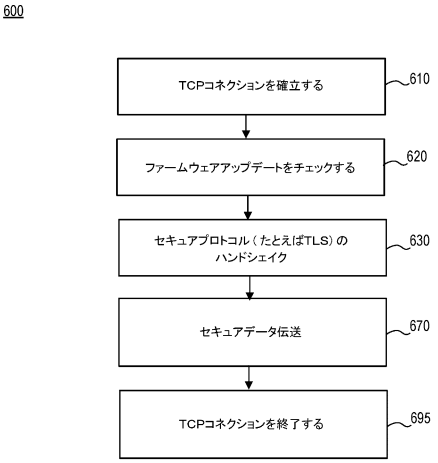
【図４】


A
B
400
450
465
457
456
アプリケーション 440
SSL/TLS 435
SMC 430
NVMデバイス 110
暗号文
TLSヘッダ
TLS OP
CMD ID
461
復号鍵
終了
AES GCM 復号
469
平文
451
IV 書き込み
TCP/IP 420
SPIドライバ 425
ホストコンピューティングシステム 102
454
455
TCPヘッダ
アプリケーション 405
SSL/TLS 410
TCP/IP 415
サーバ 105
暗号化鍵
453
AES GCM 暗号化
スタート
SW/FW
HW

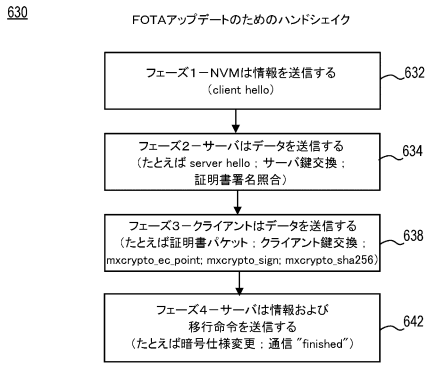
【図５】


ホストコンピューティングシステム 102
CMD ID
TLS OP
TLSヘッダ
暗号文
461
NVMデバイス 110
(SMC内の) SPI CMDデコーダ
CMD ID
125
暗号バッファ
TLS OP
TLSヘッダ
暗号文
128
マイクロコントローラ
(w/ 暗号アクセラレータ)
P/E CMD
ターゲットアドレス
長さ
ターゲットデータ
118
140
451
マイクロコントローラ
(NVMユーザアレイへの書き込み)
ターゲットアドレス
長さ
ターゲットデータ
118
ユーザアレイ(NVM)
101

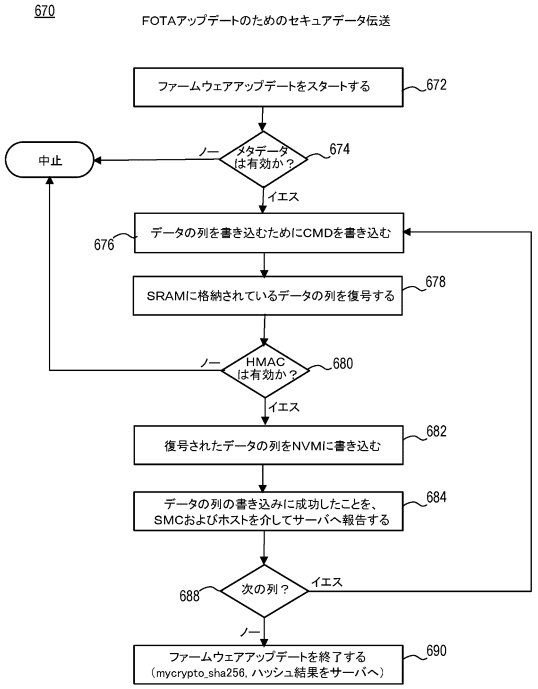
【図６】


600
TCPコネクションを確立する
610
ファームウェアアップデートをチェックする
620
セキュアプロトコル（たとえばTLS）の
ハンドシェイク
630
セキュアデータ伝送
670
TCPコネクションを終了する
695

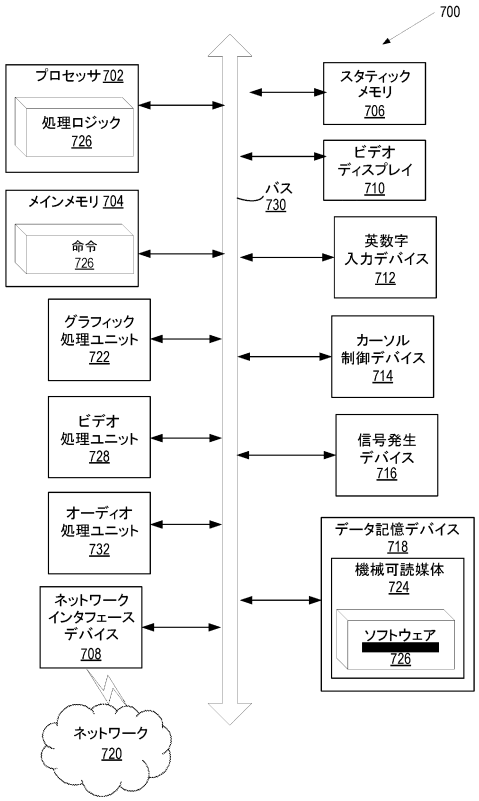
【図６Ａ】


630
FOTAアップデートのためのハンドシェイク
フェーズ1－NVMは情報を送信する
（client hello）
632
フェーズ2－サーバはデータを送信する
（たとえば server hello；サーバ鍵交換；
証明書署名照合）
634
フェーズ3－クライアントはデータを送信する
（たとえば証明書パケット；クライアント鍵交換；
mxcrypto_ec_point; mxcrypto_sign; mxcrypto_sha256）
638
フェーズ4－サーバは情報および
移行命令を送信する
（たとえば暗号仕様変更；通信 "finished"）
642

【図６Ｂ】


670
FOTAアップデートのためのセキュアデータ伝送
ファームウェアアップデートをスタートする
672
メタデータは有効か？
674
ノー
中止
イエス
データの列を書き込むためにCMDを書き込む
676
SRAMに格納されているデータの列を復号する
678
HMACは有効か？
680
ノー
イエス
復号されたデータの列をNVMに書き込む
682
データの列の書き込みに成功したことを、
SMCおよびホストを介してサーバへ報告する
684
次の列？
688
イエス
ノー
ファームウェアアップデートを終了する
（mycrypto_sha256, ハッシュ結果をサーバへ）
690

【図７】


700
プロセッサ 702
処理ロジック 726
メインメモリ 704
命令 726
グラフィック処理ユニット 722
ビデオ処理ユニット 728
オーディオ処理ユニット 732
ネットワークインタフェースデバイス 708
ネットワーク 720
バス 730
スタティックメモリ 706
ビデオディスプレイ 710
英数字入力デバイス 712
カーソル制御デバイス 714
信号発生デバイス 716
データ記憶デバイス 718
機械可読媒体 724
ソフトウェア 726

フロントページの続き

(33)優先権主張国・地域又は機関
米国(US)
早期審査対象出願
(74)代理人 100098501
弁理士 森田 拓
(74)代理人 100116403
弁理士 前川 純一
(74)代理人 100134315
弁理士 永島 秀郎
(74)代理人 100162880
弁理士 上島 類
(72)発明者 セルゲイ オストリコフ
アメリカ合衆国 カリフォルニア レッドウッドシティ チャーター ストリート ７２０
(72)発明者 スティーブン ロスナー
アメリカ合衆国 カリフォルニア キャンベル ウエスト・リンコン・アヴェニュー ５４９
(72)発明者 クリフ ジトロー
アメリカ合衆国 カリフォルニア サンノゼ カーサ・マデイラ・レーン ４３９０
審査官 松平 英
(56)参考文献 特表２００３－５１０７０６（ＪＰ，Ａ）
米国特許出願公開第２０１７／０３１０６５２（ＵＳ，Ａ１）
特表２０１９－５０２２８６（ＪＰ，Ａ）
米国特許出願公開第２０１４／０２１５１１１（ＵＳ，Ａ１）
米国特許第９３４８７７１（ＵＳ，Ｂ１）
米国特許出願公開第２０１５／０１２７９３０（ＵＳ，Ａ１）
米国特許出願公開第２００５／０１０８５７１（ＵＳ，Ａ１）
特表２００７－５１３４０６（ＪＰ，Ａ）
(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｇ０６Ｆ１２／１４
２１／００－２１／８８
Ｇ０９Ｃ １／００－５／００
Ｈ０４Ｋ １／００－３／００
Ｈ０４Ｌ ９／００－９／４０