(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6473876号
(P6473876)
(45)発行日　平成31年2月27日(2019.2.27)　(24)登録日　平成31年2月8日(2019.2.8)

(51)Int.Cl.　F I
*H04L 9/08 (2006.01)*　H04L 9/00 601C
*G06F 21/44 (2013.01)*　H04L 9/00 601E
G06F 21/44

請求項の数 10　(全 21 頁)

(21)出願番号　特願2016-244851(P2016-244851)
(22)出願日　平成28年12月1日(2016.12.1)
(65)公開番号　特開2018-93456(P2018-93456A)
(43)公開日　平成30年6月14日(2018.6.14)
審査請求日　平成30年1月25日(2018.1.25)

早期審査対象出願

(73)特許権者　519014693
株式会社ユートピア企画
東京都港区新橋六丁目２番１号木村ビル８階
(72)発明者　牧　弘之
神奈川県川崎市幸区新塚越１－２　サウザンドシティ１－３７１２
(72)発明者　余語　邦彦
東京都文京区小石川１－９－１４－２５０３

審査官　青木　重徳

最終頁に続く

(54)【発明の名称】セキュアネットワーク通信方法

(57)【特許請求の範囲】
【請求項１】
第１のネットワークで接続されるエンドポイントと中継ゲートウェイと、前記中継ゲートウェイと第２のネットワークで接続されるアプリケーションゲートウェイとを含むセキュアネットワークの通信方法であって
アプリケーションゲートウェイが、エンドポイント毎にユニークな個別識別子と認証パスワードに基づき互いに非対称な第１及び第２の認証子を生成し、第２の認証子はホワイトリストに保存して管理し、第１の認証子はエンドポイントで保存するエンドポイントの初期化段階と、
中継ゲートウェイが、エンドポイントから個別識別子と暗号化された第１の認証子から求めた演算値を含むデータを受信し、第１の認証子の正当性を判断し、正当と判断した場合は中継ゲートウェイにて生成した乱数を使用して非対称な新たな第１及び第２の認証子を生成して送信し、アプリケーションゲートウェイではホワイトリストの第２の認証子を更新し、タイムラインログを追加し、エンドポイントでは第１の認証子を更新して保存するエンドポイントの初期化時の認証段階と、
中継ゲートウェイが、エンドポイントで生成した第１の乱数とアプリケーションゲートウェイで生成した第３の乱数を受信し、中継ゲートウェイで生成した第２の乱数と第１の乱数から暗号鍵の一部を生成してアプリケーションゲートウェイに送信し、第２の乱数と第３の乱数から暗号鍵の他の一部を生成してエンドポイントに送信し、アプリケーションゲートウェイが暗号鍵の一部と第３の乱数から暗合鍵を生成してホワイトリストに保存し

、エンドポイントが暗号鍵の他の一部と第１の乱数から暗合鍵を生成して保存する共通鍵の配置段階とを有することを特徴とするセキュアネットワーク通信方法。

【請求項２】

前記中継ゲートウェイがエンドポイントから個別識別子と暗号化された最新の第１の認証子から求めた演算値を含むデータを受信し、最新の第１の認証子の正当性を判断して、正当と判断した場合は中継ゲートウェイにて新たに生成した乱数を使用して非対称な新たな第１及び第２の認証子を生成して送信し、アプリケーションゲートウェイではホワイトリストの第２の認証子を更新し、タイムラインログを追加し、エンドポイントでは第１の認証子を更新して保存する随時認証段階をさらに有することを特徴とする請求項１に記載のセキュアネットワーク通信方法。

【請求項３】

前記アプリケーションゲートウェイがエンドポイント、中継ゲートウェイ、及びアプリケーションゲートウェイのそれぞれで新たに生成した乱数を使用して次の通信用の新たな暗号鍵を生成してエンドポイントに送信し、エンドポイント、アプリケーションゲートウェイのそれぞれが暗号鍵を更新する随時鍵交換段階をさらに有することを特徴とする請求項１に記載のセキュアネットワーク通信方法。

【請求項４】

前記エンドポイントの初期化段階における第２の認証子は個別識別子のハッシュ値と認証パスワードのハッシュ値とのＸＯＲ演算値であり、第１の認証子は前記第２の認証子を予め設定された認証のためのキーであるマスターキーでエンコード処理したものであることを特徴とする請求項１に記載のセキュアネットワーク通信方法。

【請求項５】

前記エンドポイントの初期化時の認証段階において、

前記暗号化された第１の認証子から求めた演算値は、暗号化された第１の認証子のハッシュ値であり、

エンドポイントから受信するデータには、エンドポイントで生成した乱数のハッシュ値をさらに含み、

第１の認証子の正当性の判断は、個別識別子に対応してホワイトリストから抽出された第２の認証子をマスターキーでデコードした結果に対するハッシュ値と前記受信した暗号化された第１の認証子のハッシュ値から求めた演算値と、受信し乱数のハッシュ値とを比較して行うことを特徴とする請求項１に記載のセキュアネットワーク通信方法。

【請求項６】

前記エンドポイントの初期化時の認証段階における新たな第２の認証子は、中継ゲートウェイにて生成した前記乱数と個別識別子に対応してホワイトリストから抽出された第２の認証子とのＸＯＲ演算値であり、新たな第１の認証子は、前記新たな第２の認証子をマスターキーでエンコード処理したものであることを特徴とする請求項１または５に記載のセキュアネットワーク通信方法。

【請求項７】

前記共通鍵の配置段階における前記暗号鍵の一部は、第１の乱数と第２の乱数とのＸＯＲ演算値であり、暗号鍵の他の一部は第２の乱数と第３の乱数とのＸＯＲ演算値であり、アプリケーションゲートウェイが生成する暗合鍵とエンドポイントが生成する暗合鍵はいずれも第１の乱数と第２の乱数と第３の乱数とのＸＯＲ演算値であることを特徴とする請求項１に記載のセキュアネットワーク通信方法。

【請求項８】

前記エンドポイントで生成または取得したデータを含むペイロードを、前記中継ゲートウェイを介して前記アプリケーションゲートウェイに送信する段階をさらに含み、

前記ペイロードは、個別識別子、機能を示すファンクションＩＤ、暗号鍵、及びデータの４つの枠で構成され、

ペイロードのデータ枠には前記送信されるデータを最新の暗号鍵でエンコード処理した値が格納され、

送信されたデータは、アプリケーションゲートウェイにて最新の暗号鍵でデコード処理することにより復号されることを特徴とする請求項１乃至３に記載のセキュアネットワーク通信方法。

【請求項９】

前記第１のネットワークは無線によるローカルネットワークであり、前記第２のネットワークはＴＣＰによるインターネットであることを特徴とする請求項１に記載のセキュアネットワーク通信方法。

【請求項１０】

前記タイムラインログの記録は個別識別子、タイムスタンプ、及び第２の認証子を認証パスワードのハッシュ値でエンコードした値をタイムラインログ記録用のメモリに保存することで行われ、

前記タイムラインログの追加は個別識別子、新たなタイムスタンプ、及び最新の第２の認証子を認証パスワードのハッシュ値でエンコードした値をタイムラインログ記録用のメモリに追加して保存することで行われ、

第１の認証子の初期値は、タイムラインログに記録された第２の認証子を認証パスワードのハッシュ値でエンコードした値を用いて、第２の認証子を順次デコードすることで順次更新前の第２の認証子にさかのぼり、最後に得られた第２の認証子の初期値をマスターキーでデコード処理することで求められることを特徴とする請求項１に記載のセキュアネットワーク通信方法。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、セキュアネットワーク通信方法に関し、特にデータの発生源でデータを捕捉する機器をホワイトリストに登録して管理するとともに起動時の認証の記録をタイムライン管理によりライフサイクルで管理し、データの発生源からデータベースの入り口までを暗号通信を行うことができるセキュアネットワーク通信方法に関する。

【背景技術】

【０００２】

ビッグデータのデータの発生源は、センサーのような定置に置かれるような装置から捕捉される。そのようなデータ発生源からデータを集積する場合、データ発生源と集積されるデータベースの間には、複数の通信手段、ゲートウェイ、ルータ、ファイルシステムなどが使われる。なかでも、末端のデバイスから、インターネットに中継するためには、ＺｉｇＢｅｅ（登録商標）、Ｂｌｕｅｔｏｏｔｈ（登録商標）、ＷｉＦｉ（登録商標）のような無線によるローカルエリアネットワークと、ＴＣＰによるインターネットの通信という異なった通信手段とセキュリティ技術を組み合わせたゲートウェイが必要である。

【０００３】

インターネットの発展により、その通信に対するセキュリティはＳＳＬ／ＴＬＳのように標準化され、確立された体系がある。しかしながら、データ発生源に置かれるデバイスは直接インターネットに接続されるのではなく、もっと運用コストが抑えられ、最後の数十メートルのネットワークを構成するのに都合のよいＺｉｇＢｅｅ、Ｂｌｕｅｔｏｏｔｈのような無線による限られたエリアのネットワークが使われている。このエリアは必ずしも安定した電力が供給されたり、制御されたりして一定の環境が維持されるわけでもない場合が多い。そのような環境に置かれるデバイスにはできる限り省電力で負荷を減らすこと、コストを抑えることという条件が求められる。

【０００４】

インターネットの安全性のためのセキュリティ技術は、このエリアへの適用が予定されているとは言えない。例えば、キー長による安全性と通信する機器に使用されるＬＳＩの性能というようなトレードオフの問題もあり、充分なセキュリィと言われる少なくとも１２８ビット以上の鍵長を実装することはこのエリアに置かれるデバイスで使われるＬＳＩの性能を規定する価格要件には、まだ相当な課題がある。また、ＳＳＬのような暗号方式

では、証明書を事前にインストールするというような手続きと、鍵配送時に必要とされる計算能力も課題の一つと考えられる。そういった背景から、データ発生源からデータベースに接続するという発想のより手軽なデータ連携のためのネットワークと暗号方式が求められる。

【０００５】

センサーから得られるデータは短く、それほどリアルタイム性が要求されないものが殆どであり、そもそも暗号化の必要のないものが多い。しかしながら、成りすましによって異常なデータで攻撃された場合、仕組みとして混乱し大きな被害が予想される。つまり、セキュリティの要件としてはデータの暗号化も一つではあるが、機器認証、成りすましの対策が相対的に重要であると言える。

【０００６】

特許文献１は、第１装置と第２装置とが中継装置を介して共通の暗号鍵を共有する技術を開示したものであるが、データの発生源でデータを捕捉するような機器の認証に関しての記載は無い。

特許文献２は、アドホックネットワーク内のノードが用いる暗号鍵はゲートウェイごとに共通とし、交換が必要になるとゲートウェイで新鍵を生成するネットワークに関する提案であるが、個々のノードの鍵交換は別途携帯端末を接続して管理サーバーから入手するなど管理が容易ではない。

高性能のＬＳＩの組み込みや高機能の実装が期待できないセンサーのようなデータを捕捉する機器でもセキュリティの確保された通信が行えるシステムの提供が望まれる。

【先行技術文献】

【特許文献】

【０００７】

【特許文献１】特許第５０３９１４６号公報

【特許文献２】特許第５４８８７１６号公報

【発明の概要】

【発明が解決しようとする課題】

【０００８】

本発明は、上記従来のセキュアネットワーク通信方法における問題点に鑑みてなされたものであって、本発明の目的は、データの発生源でデータを捕捉する機器を独自の非対称の認証子を有するネットワークのエンドポイントとして事前にホワイトリストに登録して管理し、エンドポイントが起動時に認証され、その認証の記録をタイムライン管理によりライフサイクルで管理し、データの発生源からデータベースの入り口までを通信手段に影響されることなくエンド・エンドで暗号通信を行うことができるセキュアネットワーク通信方法を提供することにある。

【課題を解決するための手段】

【０００９】

上記目的を達成するためになされた本発明によるセキュアネットワーク通信方法は、第１のネットワークで接続されるエンドポイントと中継ゲートウェイと、前記中継ゲートウェイと第２のネットワークで接続されるアプリケーションゲートウェイとを含むセキュアネットワークの通信方法であって、アプリケーションゲートウェイが、エンドポイント毎にユニークな個別識別子と認証パスワードに基づき互いに非対称な第１及び第２の認証子を生成し、第２の認証子はホワイトリストに保存して管理するとともにタイムラインログに記録し、第１の認証子はエンドポイントで保存するエンドポイントの初期化段階と、中継ゲートウェイが、エンドポイントから個別識別子と暗号化された第１の認証子から求めた演算値を含むデータを受信し、第１の認証子の正当性を判断し、正当と判断した場合は中継ゲートウェイにて生成した乱数を使用して非対称な新たな第１及び第２の認証子を生成して送信し、アプリケーションゲートウェイではホワイトリストの第２の認証子を更新し、タイムラインログを追加し、エンドポイントでは第１の認証子を更新して保存するエンドポイントの初期化時の認証段階と、中継ゲートウェイが、エンドポイントで生成した

第１の乱数とアプリケーションゲートウェイで生成した第３の乱数を受信し、中継ゲートウェイで生成した第２の乱数と第１の乱数から暗号鍵の一部を生成してアプリケーションゲートウェイに送信し、第２の乱数と第３の乱数から暗号鍵の他の一部を生成してエンドポイントに送信し、アプリケーションゲートウェイが暗号鍵の一部と第３の乱数から暗合鍵を生成してホワイトリストに保存し、エンドポイントが暗号鍵の他の一部と第１の乱数から暗合鍵を生成して保存する共通鍵の配置段階とを有することを特徴とする。

【００１０】

前記中継ゲートウェイがエンドポイントから個別識別子と暗号化された最新の第１の認証子から求めた演算値を含むデータを受信し、最新の第１の認証子の正当性を判断して、正当と判断した場合は中継ゲートウェイにて新たに生成した乱数を使用して非対称な新たな第１及び第２の認証子を生成して送信し、アプリケーションゲートウェイではホワイトリストの第２の認証子を更新し、タイムラインログを追加し、エンドポイントでは第１の認証子を更新して保存する随時認証段階をさらに有することが好ましい。

【００１１】

前記アプリケーションゲートウェイがエンドポイント、中継ゲートウェイ、及びアプリケーションゲートウェイのそれぞれで新たに生成した乱数を使用して次の通信用の新たな暗号鍵を生成してエンドポイントに送信し、エンドポイント、アプリケーションゲートウェイのそれぞれが暗号鍵を更新する随時鍵交換段階をさらに有することが好ましい。

【００１２】

前記エンドポイントの初期化段階における第２の認証子は個別識別子のハッシュ値と認証パスワードのハッシュ値とのＸＯＲ演算値であり、第１の認証子は前記第２の認証子を予め設定された認証のためのキーであるマスターキーでエンコード処理したものであることが好ましい。

【００１３】

前記エンドポイントの初期化時の認証段階において、前記暗号化された第１の認証子から求めた演算値は、暗号化された第１の認証子のハッシュ値であり、エンドポイントから受信するデータには、エンドポイントで生成した乱数のハッシュ値をさらに含み、第１の認証子の正当性の判断は、個別識別子に対応してホワイトリストから抽出された第２の認証子をマスターキーでデコードした結果に対するハッシュ値と前記受信した暗号化された第１の認証子のハッシュ値から求めた演算値と、受信し乱数のハッシュ値とを比較して行うことが好ましい。

【００１４】

前記エンドポイントの初期化時の認証段階における新たな第２の認証子は、中継ゲートウェイにて生成した前記乱数と個別識別子に対応してホワイトリストから抽出された第２の認証子とのＸＯＲ演算値であり、新たな第１の認証子は、前記新たな第２の認証子をマスターキーでエンコード処理したものであることが好ましい。

【００１５】

前記共通鍵の配置段階における前記暗号鍵の一部は、第１の乱数と第２の乱数とのＸＯＲ演算値であり、暗号鍵の他の一部は第２の乱数と第３の乱数とのＸＯＲ演算値であり、アプリケーションゲートウェイが生成する暗合鍵とエンドポイントが生成する暗合鍵はいずれも第１の乱数と第２の乱数と第３の乱数とのＸＯＲ演算値であることが好ましい。

【００１６】

前記エンドポイントで生成または取得したデータを含むペイロードを、前記中継ゲートウェイを介して前記アプリケーションゲートウェイに送信する段階をさらに含み、　前記ペイロードは、個別識別子、機能を示すファンクションＩＤ、暗号鍵、及びデータの４つの枠で構成され、ペイロードのデータ枠には前記送信されるデータを最新の暗号鍵でエンコード処理した値が格納され、送信されたデータは、アプリケーションゲートウェイにて最新の暗号鍵でデコード処理することにより復号されることが好ましい。

前記第１のネットワークは無線によるローカルネットワークであり、前記第２のネットワークはＴＣＰによるインターネットであることが好ましい。

【００１７】
前記タイムラインログの記録は個別識別子、タイムスタンプ、及び第２の認証子を認証パスワードのハッシュ値でエンコードした値をタイムラインログ記録用のメモリに保存することで行われ、前記タイムラインログの追加は個別識別子、新たなタイムスタンプ、及び最新の第２の認証子を認証パスワードのハッシュ値でエンコードした値をタイムラインログ記録用のメモリに追加して保存することで行われ、第１の認証子の初期値は、タイムラインログに記録された第２の認証子を認証パスワードのハッシュ値でエンコードした値を用いて、第２の認証子を順次デコードすることで順次更新前の第２の認証子にさかのぼり、最後に得られた第２の認証子の初期値をマスターキーでデコード処理することで求められることが好ましい。

【発明の効果】
【００１８】
本発明によるセキュアネットワーク通信方法によれば、エンドポイントの認証子は認証の行われる毎に、通信の両端にあるエンドポイントとアプリケーションゲートウェイとは独立した別の装置で新たに生成した乱数により更新され、エンドポイントとアプリケーションゲートウェイとでは異なるものとして保存され、又更新の履歴がタイムラインにより初期状態まで遡及できるため、途中から第３者がなりすましにより通信に割り込むことが極めて困難な通信環境を提供することができる。
また本発明によるセキュアネットワーク通信方法によれば、エンドポイントはホワイトリストにより管理するため、ブラックリスト管理の様に、リスト内容が事あるごとに増え続ける心配がなくアプリケーションゲートウェイの管理負荷を不必要に増やすおそれがない。

【図面の簡単な説明】
【００１９】
【図１】本発明の実施形態によるセキュアネットワーク通信方法を適用するネットワーク構成要素とデータの連係を概略的に示す図である。
【図２】本発明の実施形態によるエンドポイント、中継ゲートウェイ、アプリケーションゲートウェイによる機能構成を示す図である。
【図３】本発明の実施形態によるゲートウェイと関連するアプリケーションの構成例を示す図である。
【図４】本発明の実施形態によるセキュアネットワーク通信方法におけるエンドポイントの初期化の方法を示す図である。
【図５】本発明の実施形態によるセキュアネットワーク通信方法におけるエンドポイントの初期化時の認証方法を示す図である。
【図６】本発明の実施形態によるセキュアネットワーク通信方法におけるエンドポイントのアテスト時の認証方法を示す図である。
【図７】本発明の実施形態によるタイムライン管理方法を示す図である。
【図８】本発明の実施形態による中継ゲートウェイを経由したエンド・エンドでの共通鍵の配置方法を示す図である。
【図９】本発明の実施形態による中継ゲートウェイを経由したエンド・エンドでの暗号通信の鍵の交換方法を示す図である。
【図１０】本発明の実施形態による中継ゲートウェイを経由したエンド・エンドでの暗号通信の鍵の交換とデータの送受信方法を示す図である。

【発明を実施するための形態】
【００２０】
次に、本発明に係るセキュアネットワーク通信方法を実施するための形態の具体例を、図面を参照しながら説明する。
図１は、本発明の実施形態によるセキュアネットワーク通信方法を適用するネットワーク構成要素とデータの連係を概略的に示す図である。
【００２１】

図１を参照するとセキュアネットワークの構成要素として、エンドポイント１０、中継ゲートウェイ２０、アプリケーションゲートウェイ３０を含む。エンドポイント１０は温度や圧力などを検出する各種のセンサーやこれらセンサーを取りまとめる端末機器でもよく、人が操作するハンディーターミナルでもスマートフォンのような携帯機器でもよい。ここでエンドポイントはデータの発生源でデータを捕捉する機器の総称であり、データ送信の起点となる機器を表す。エンドポイント１０はセキュア通信における認証子の管理や暗号・復号などを実行するためのインターフェースであるエンドポイントＡＰＩ１１を備える。

【００２２】

アプリケーションゲートウェイ３０は、エンドポイントからの各種データを受信し、最終的にそのデータの処理を行うアソシエートアプリケーションにデータを引き渡す役割を果たす。ここでアソシエートアプリケーションとは、例えばエンドポイントで補足したデータに基づき特定エリアの温度分布を提供したり、時系列変化から温度変化を予測したりするソフトウェアのように、データを利用して関連する情報を生成して提供する実務的なソフトウェアを指す。

【００２３】

アプリケーションゲートウェイ３０は、データの受け渡しや暗号・復号、乱数生成などを実行するためのインターフェースであるアプリケーションＡＰＩ３１を備える。アプリケーションＡＰＩ３１は上記機能の他、ネットワークの監視やエンドポイントの管理のために作成するホワイトリストの管理なども行う。さらにアプリケーションＡＰＩ３１は、通信の際の認証に使用する認証子の更新の履歴をログに残し、不正アクセスを検知したりログ情報を照会したりするタイムライン管理も行う。アプリケーションＡＰＩ３１が管理する情報はネットワーク情報管理メモリ３３に保存される。またアプリケーションＡＰＩ３１を介して最終的にデータが引き渡されるアソシエートアプリケーションはアソシエートアプリＤＢ３２に保存される。

【００２４】

アプリケーションゲートウェイ３０、アプリケーションＡＰＩ３１、ネットワーク情報管理メモリ３３、及びアソシエートアプリＤＢ３２は１つのデータセンターとして具現化してもよく、またネットワーク情報管理メモリ３３、及びアソシエートアプリＤＢ３２はアプリケーションゲートウェイ３０に直結される外部メモリとして構成されてもよい。

【００２５】

中継ゲートウェイ２０は、エンドポイント１０からデータを受信してアプリケーションゲートウェイ３０にデータを引き渡す。エンドポイント１０と中継ゲートウェイ２０とはＰＡＮ（Ｐｅｒｓｏｎａｌ　Ａｒｅａ　Ｎｅｔｗｏｒｋ）４０によって接続される。通信方法としてはＧｉｇｂｅｅ、Ｂｌｕｅｔｏｏｔｈ、Ｗｉｆｉなどが適用可能であり、データの内容やそれぞれの無線通信の特性によって使い分けてもよい。また図１に示すようにエンドポイント１０と中継ゲートウェイ２０の間に複数のノードを介してもよい。

【００２６】

一方中継ゲートウェイ２０とアプリケーションゲートウェイ３０との間はインターネット５０を介して接続される。インターネットの安全性に関してはすでにＳＳＬ／ＴＳＬのような暗号通信をベースとする通信環境が整っており、本明細書においても中継ゲートウェイ２０とアプリケーションゲートウェイ３０との間はこうした既存の通信技術にて安全な通信が確保されることを前提とし、インターネットの５０の安全性に関する記載は省略する。

【００２７】

エンドポイントで補足されるデータは、所定の処理を行うために使用されるものであり、どのような機能で、どのような処理を行うデータであるかという意味を含むファンクションと結び付けられる。ファンクションはデータ連携の通信を行う起点の前と終点の先にある機能である。エンドポイント１０は送信するデータをそのデータのファンクションを表すファンクションＩＤと連携して扱うことができる。ファンクションとはエンドポイン

トＡＰＩ１１とアプリケーションＡＰＩ３１の間で論理的なデータ連携の単位になり、エンドポイントＡＰＩ１１とアプリケーションＡＰＩ３１のそれぞれで独立した単位の処理として扱うことができる。

【００２８】

エンドポイント１０側のファンクションは、センサーなどから発生したデータをデジタル化して送る単位であり、エンドポイントＡＰＩ１１はそのデータを、ファンクションＩＤを付けて送信する機能を持つ。

アプリケーションＡＰＩ３１側のファンクションは、センサーなどからのデータをデータベースに蓄積するために、データをファンクションＩＤとともに受け取り、ファンクションＩＤに関連付けられたＤＬＬ（Ｄｙｎａｍｉｃ　Ｌｉｎｋ　Ｌｉｂｒａｒｙ）をダイナミックに接続する機能を持ち、ファンクションＩＤに対応付けられた、ＤＬＬ内のメソッドを呼び出し、そのメソッドに関連付けられたデータベースに接続し、メソッドに記述されたプロシージャ（関数）が最終的に呼ばれるＡＰＩで扱われる単位である。

【００２９】

本発明によるセキュアネットワーク通信方法は、こうした構成要素からなるネットワークにおいて、エンドポイントの初期化段階にアプリケーションゲートウェイが、エンドポイント毎にユニークな個別識別子と認証パスワードに基づき互いに非対称な第１及び第２の認証子を生成し、第２の認証子はホワイトリストに保存して管理し、第１の認証子はエンドポイントで保存し、エンドポイントの認証を行う毎に、新たに生成する乱数を使用して互いに非対称な認証子を生成してホワイトリストを更新するとともに、タイムラインログに記録して認証子のユニーク性を時系列的に管理し、データの送信に関してはエンドポイント、中継ゲートウェイ、アプリケーションゲートウェイのそれぞれが生成する乱数から生成される暗号鍵を次の通信の共通の暗号鍵としてエンドポイントとアプリケーションゲートウェイで交換することにより、なりすましを防止し、比較的小さな負荷で安全な通信方法を提供するものである。

【００３０】

図２は本発明の実施形態によるエンドポイント、中継ゲートウェイ、アプリケーションゲートウェイによる機能構成を示す図である。図２はそれぞれの構成要素が持つデータ通信に関する機能とその内容を示す。

【００３１】

図２を参照すると、アプリケーションゲートウェイは通信、セッション管理、アプリケーション管理、ＡＰＩ、及びファンクションの機能を有する。通信は対岸である中継ゲートウェイとのインターネット上の通信のためのプロトコルＴＣＰ／ＩＰの管理などを行う。セッション管理は中継ゲートウェイ経由でエンドポイントからデータを受信する通信の通信セッション管理を行う機能である。アプリケーション管理は、受信したデータを受け渡し先のアプリケーションに中継する機能である。ＡＰＩは受信したデータを復号化し、データに付加されたファンクションＩＤを参照し、ファンクションＩＤに対応付けられたＤＬＬを接続し、関連するアソシエートアプリケーションにデータを引き渡す。こうしたデータの受け渡しはファンクションごとに行われるため、図２に示すようにファンクションは、その数に応じて複数の機能（ファンクション１、ファンクション２、・・・）を有する。

【００３２】

中継ゲートウェイは、データＢＵＳを挟んでアプリケーションゲートウェイ側のインターネット通信のための通信、及びセキュリティ管理とエンドポイント側のＰＡＮによる通信のための通信、セッション管理、セキュリティ管理の機能を有する。エンドポイント側の通信は、無線通信の方式（ＺｉｇＢｅｅ、Ｂｌｕｅｔｏｏｔｈ、ＷｉＦｉなど）に応じてそれぞれの規格に準拠した通信の管理を行う。セキュリティ管理ではデータの中継の機能の他に、エンドポイント側のセキュリティ管理に関し、エンドポイントの認証の機能を有する。またデータの送受信のための暗号鍵を生成して交換するための乱数を生成する機能も備える。

【００３３】
エンドポイントは、中継ゲートウェイとの通信、通信管理、ＡＰＩ、及びファンクションの機能を有する。エンドポイントはデータ送信の起点であり、ＡＰＩではこうしたデータ発生源から補足したデータの暗号化の機能を有する。また次の通信に使用する暗号鍵の生成及び更新に必要な暗号を生成する機能や、アプリケーションゲートウェイから送信される暗号化された暗号鍵の一部を復号化する機能も備える。エンドポイントで扱うデータはファンクションを伴い、ファンクションが複数ある場合は、その数に応じて複数の機能（ファンクション１、ファンクション２、・・・）を有する。ＡＰＩの上記の機能はファンクション毎に設定することができ、またエンドポイントでのファンクションと、アプリケーションゲートウェイのファンクションとは１対１に対応付けることができ、したがって、ファンクションＩＤ別にデータ連携を行うことができる。

【００３４】
図３は、本発明の実施形態によるゲートウェイと関連するアプリケーションの構成例を示す図である。
アプリケーションゲートウェイ３０は通信データのデータ連携を管理する中で、接続の受信処理を行うためにメインスレッドを起動し、ＴＣＰ／ＩＰデータ受信サーバの機能を果たすアプリケーションを立ち上げる（段階Ｓ３１０）。リスナーの場合リスナーソケットを作成し、ソケットはＴＣＰポートを指定しＩＰアドレスに対しＴＣＰ接続が処理されるように設定することができる。

【００３５】
次にデータの読み込みを行うために１件の送受信処理スレッドを起動する（段階Ｓ３２０）。送受信処理スレッドでは受信データの読込みの他、データに付加されたファンクションＩＤに関連付けられるＤＬＬであるデータ連携管理．ＤＬＬのダイナミックロードやファンクションＩＤから関連メソッドの呼び出しなどを行う（段階Ｓ３３０）。

【００３６】
データ連携管理．ＤＬＬは、データ連携管理に利用されるライブラリで、管理モニタ画面の制御、ホワイトリスト管理、ファンクション連携、暗号／復号などの処理に利用される。データ連携管理．ＤＬＬが処理するホワイトリスト、タイムラインログ、エンドポイントの状況などのセキュリティに関する情報は、ネットワーク情報管理メモリに保存され、処理の内容に応じてネットワーク情報管理メモリから読み出したり、ネットワーク情報管理メモリに書き込んだりして、最新情報が保存されるようにする。
またエンドポイントから受信したデータは、ファンクションＩＤから呼び出されるメソッドに記述されたプロシージャ（関数）に引き当てられ、最終的に得られたデータがアソシエートアプリＤＢの対応するデータベースに保存される。

【００３７】
次に図４～１０を用いて本発明の実施形態によるセキュアネットワーク通信方法の具体的な処理手順について説明する。
図４は本発明の実施形態によるセキュアネットワーク通信方法におけるエンドポイントの初期化の方法を示す図である。
エンドポイントの初期化は、アプリケーションゲートウェイにて、エンドポイントの初期化時の認証に使用する最初の非対称の認証子、即ち認証子の初期値（$C_0$，$A_0$）を発番してホワイトリストに格納する処理である。

【００３８】
この段階では、アプリケーションゲートウェイが、エンドポイント毎にユニークな個別識別子と認証パスワードに基づき互いに非対称な第１及び第２の認証子を生成し、第２の認証子はホワイトリストに保存して管理し、第１の認証子はエンドポイントで保存する。
図４を参照すると、新たに設置するエンドポイントの初期化に当たり、エンドポイント毎に固有の個別識別子であるＥＰＩＤとＥＰＩＤの認証を許可するための認証パスワードを入力する。個別識別子ＥＰＩＤはエンドポイントにネイティブであって、かつユニーク性を表す識別子であることが好ましく、例えばＩＥＥＥ８０２アドレス等が利用できる。

また認証パスワードはＥＰＩＤ毎に固有のものであることが好ましく、事前に他のエンドポイントの認証パスワードと重複しないように設定しておく。

【００３９】

アプリケーションゲートウェイは、アプリケーションＡＰＩを使用して、以下の処理を行う。

まず、入力された個別識別子ＥＰＩＤのハッシュ（Ｈａｓｈ）値をもとめ，非対称の認証子の内の第２の認証子の初期値$Ｃ_{0}$を生成する前段階の$Ｃ_{-1}$に代入する。次に認証パスワードのハッシュ値をもとめこの値を$Ｚ_{0}$とする。このように認証パスワードもハッシュ関数を通すことにより、元の認証パスワードを推定できない形に変換してから使用する。ここで前提として個別識別子ＥＰＩＤと認証パスワードのそれぞれのハッシュ関数で生成されるダイジェスト値は、例えば１２８ビットのように同じ長さであるとする。

【００４０】

上記で求めた$Ｃ_{-1}$と$Ｚ_{0}$とのＸＯＲ演算を行い、その結果を第２の認証子の初期値$Ｃ_{0}$とする。

第２の認証子の初期値$Ｃ_{0}$は、個別識別子ＥＰＩＤとともにホワイトリストとしてエンドポイント管理テーブルに登録する。エンドポイント管理テーブルは、ネットワーク情報管理メモリの中に作成されて管理されるテーブルである。

【００４１】

最後に第２の認証子の初期値$Ｃ_{0}$は、予めネットワークシステムに定められた認証のためのキーであるマスターキーでエンコード処理され、その結果として第１の認証子の初期値$Ａ_{0}$が求められる。この第１の認証子の初期値$Ａ_{0}$は出力してエンドポイントのメモリに保存する。この場合、電源を切断してもデータが消失しないＥＥＰＲＯＭやＳＩＭなどのメモリを使用する。これによりエンドポイントの認証子の初期値は消失することなく保存されるので、意図せずに電源が遮断されることが起こっても、電源を再投入すれば、エンドポイントの初期化時の認証を行うことができる。

【００４２】

エンコード処理に使用する関数は、

Ｙ＝Ｅｎｃｏｄｅ（Ｘ，ＰＷ）・・・（１）

Ｘ＝Ｄｅｃｏｄｅ（Ｙ，ＰＷ）・・・（２）

ここでＰＷはパスワード

の関係を満たすような関数であり、パスワードでエンコード処理した結果を、同じパスワードでデコード処理すると元の値に戻る性質を持つ関数である。

【００４３】

以上のように、１つの認証子の初期値$Ｃ_{0}$をもとに、マスターキーをパスワードとしてエンコード処理をすることにより、元の値$Ｃ_{0}$とは非対称な認証子の初期値$Ａ_{0}$を生成する。エンドポイントの初期化段階において、生成した非対称の認証子の初期値の内、第１の認証子の初期値$Ａ_{0}$はエンドポイントで保存され、第２の認証子の初期値$Ｃ_{0}$はアプリケーションゲートウェイに保存される。また認証パスワードそのものはアプリケーションゲートウェイのデータベースには残さないようにする。このように関連する情報を、形を変えたり別々に保存したりすることにより、ネットワーク通信の安全性を向上することができる。

【００４４】

図５は、本発明の実施形態によるセキュアネットワーク通信方法におけるエンドポイントの初期化時の認証方法を示す図であり、中継ゲートウェイの下に、新たに１つのエンドポイントを追加するときの認証の流れを示す図である。

この段階では、中継ゲートウェイが、エンドポイントから個別識別子と暗号化された第１の認証子から求めた演算値を含むデータを受信し、第１の認証子の正当性を判断し、正当と判断した場合は中継ゲートウェイにて生成した乱数を使用して非対称な新たな第１及び第２の認証子を生成して送信し、アプリケーションゲートウェイではホワイトリストの第２の認証子を更新し、タイムラインログを追加し、エンドポイントでは第１の認証子を

更新して保存する。

【００４５】

前提として図４で説明したエンドポイントの初期化段階が終了しており、エンドポイントの不揮発性メモリには第１の認証子の初期値$A_0$が保存され、アプリケーションゲートウェイのエンドポイント管理テーブルにはエンドポイントの固有の個別識別子であるＥＰＩＤと第２の認証子の初期値$C_0$が保存されている。したがって、エンドポイントの処理にて現在の認証子を表すｃｕｒｒｅｎｔＡは初期値$A_0$であり、エンドポイント管理テーブルでは現在の認証子を表すｃｕｒｒｅｎｔＣは初期値$C_0$である。

また、この段階はエンドポイント、中継ゲートウェイ、アプリケーションゲートウェイの３つの構成要素間の通信が行われるが、中継ゲートウェイとアプリケーションゲートウェイとの間の通信はインターネットによる暗号通信が確立されていることを前提とする。

【００４６】

図５を参照すると、最初にエンドポイントにて新たな乱数$X_0$を生成する。この乱数の生成には疑似乱数生成器を使用してもよい。

エンドポイントの初期化時の認証段階でエンドポイントから中継ゲートウェイに送信されるデータのペイロード（Ｄ１）は認証子（Ａ）とハッシュ（Ｈ）により構成される。

エンドポイントでは上記で生成した乱数$X_0$を使用して、第１の認証子の初期値$A_0$のハッシュ値と乱数$X_0$とのＸＯＲ演算を行い、結果をペイロードの認証子（Ａ）の値（Ｄ１．Ａ）に引き当てる。続いて乱数$X_0$のハッシュ値を求めペイロードのハッシュ（Ｈ）の値（Ｄ１．Ｈ）に引き当てる。

エンドポイントは、ペイロード（Ｄ１）を個別識別子ＥＰＩＤと共に中継ゲートウェイに送信してアテスト要求を行う。

【００４７】

中継ゲートウェイでは、ペイロード（Ｄ１）と個別認証子ＥＰＩＤを受信すると、ＥＰＩＤに基づいて、アプリケーションゲートウェイからマスターキーと第２の認証子の初期値$C_0$を取得して、第２の認証子の初期値$C_0$をマスターキーでエンコード処理した結果のハッシュ値を求め（演算値１；ｔｅｍｐ１）、エンドポイントから受信したペイロードの認証子Ａの値Ｄ１．Ａと演算値１とのＸＯＲ演算を行い演算値２；ｔｅｍｐ２を求める。

【００４８】

この処理において、第２の認証子の初期値$C_0$をマスターキーでエンコード処理するのは、図４で説明した第１の認証子の初期値$A_0$を求める処理と同じであり、演算値１は第１の認証子の初期値$A_0$のハッシュ値に相当する。そこで演算値２は、第１の認証子の初期値$A_0$のハッシュ値と乱数$X_0$とのＸＯＲ演算結果に、更に第１の認証子の初期値$A_0$のハッシュ値をＸＯＲ演算したものということになるので、エンドポイントの現在の認証子$A_0$が改ざんされていなければ、結果的に乱数$X_0$と同等となる。

【００４９】

中継ゲートウェイは、更に演算値２のハッシュ値を求め（演算値３；ｔｅｍｐ３）、エンドポイントから受信したペイロードのハッシュＨの値Ｄ１．Ｈと演算値３との比較によりアテストを行う。この値は、エンドポイントの現在の認証子$A_0$が改ざんされていなければ、ともに乱数$X_0$のハッシュ値となるのでアテストは成功となるが、改ざんが行われていると演算値２が乱数$X_0$と同等ではなくなるためアテストは不成功となる。

【００５０】

アテストが成功すると、中継ゲートウェイは新たな乱数$Y_0$を生成し、乱数$Y_0$と第２の認証子の初期値$C_0$とのＸＯＲ演算を行い、新たな第２の認証子$C_1$を生成する。さらに、新たな第２の認証子$C_1$をマスターキーでエンコード処理を行い新たな第１の認証子$A_1$を生成する。中継ゲートウェイで生成される乱数$Y_0$は、非決定論型の乱数生成器によって生成される。したがって乱数$Y_0$は予測不可能な値である。

【００５１】

中継ゲートウェイは、最後に新たな第２の認証子$C_1$をアプリケーションゲートウェイ

に送信し、新たな第１の認証子$A_1$は上記の演算値１とＸＯＲ演算を行った後、エンドポイントに送信する。このようにすることによって新たな第１の認証子$A_1$が通信経路にそのままの形で露出するのを防ぐことができる。

【００５２】

中継ゲートウェイから新たな第２の認証子$C_1$を受信したアプリケーションゲートウェイでは、ホワイトリストの管理としてエンドポイント管理テーブルの現在の認証子を表すｃｕｒｒｅｎｔＣを初期値$C_0$から新たな第２の認証子$C_1$に更新する。この時、アプリケーションゲートウェイは、第２の認証子の初期値$C_0$を第２の認証子$C_1$でエンコード処理を行い、この結果をタイムラインＩＤの初期値$T_0$として個別識別子ＥＰＩＤ、タイムスタンプと共にタイムラインログに記録する。

【００５３】

一方、中継ゲートウェイから新たな第１の認証子$A_1$と演算値１とのＸＯＲ演算結果を受信したエンドポイントは、受信した値と、エンドポイントが保存する第１の認証子の初期値$A_0$のハッシュ値とのＸＯＲ演算を行う。演算値１は上記のように第１の認証子の初期値$A_0$のハッシュ値を求めた結果に相当するのでこの演算結果は、新たな第１の認証子$A_1$を求めることに他ならない。こうして得られた新たな第１の認証子$A_1$で、現在の認証子を表すｃｕｒｒｅｎｔＡに記録された初期値$A_0$を更新する。

【００５４】

図６は、本発明の実施形態によるセキュアネットワーク通信方法におけるエンドポイントのアテスト時の認証方法を示す図である。ここでのアテストはエンドポイントのリブート時や非定期に行う場合であり、その際の認証の流れを示す。

この段階では中継ゲートウェイがエンドポイントから個別識別子と暗号化された最新の第１の認証子から求めた演算値を含むデータを受信し、最新の第１の認証子の正当性を判断して、正当と判断した場合は中継ゲートウェイにて新たに生成した乱数を使用して非対称な新たな第１及び第２の認証子を生成して送信し、アプリケーションゲートウェイではホワイトリストの第２の認証子を更新し、タイムラインログを追加し、エンドポイントでは第１の認証子を更新して保存する。

【００５５】

基本的な流れは図５で説明したエンドポイントの初期化時の認証と同様である。説明を一般化するため自然数ｎを添え字として用いている。

即ちアテスト時の認証の前提として、エンドポイントにおいては、現在の認証子を表すｃｕｒｒｅｎｔＡには最新の第１の認証子$A_n$が記録され、エンドポイント管理テーブルでは、現在の認証子を表すｃｕｒｒｅｎｔＣは最新の第２の認証子$C_n$が記録されている。また、タイムラインログには認証子の更新が行われる毎に、個別識別子ＥＰＩＤ、タイムスタンプ、及びタイムラインＩＤが追加されて記録され、最後に記録されたタイムラインログには、最新の第２の認証子$C_n$の更新前の第２の認証子$C_{n-1}$を最新の第２の認証子$C_n$でエンコード処理を行ってもとめたタイムラインＩＤである$T_{n-1}$が記録されている。

【００５６】

アテストが必要となると、エンドポイントでは新たな乱数$X_n$を生成し、現在の第１の認証子$A_n$のハッシュ値と乱数$X_n$とのＸＯＲ演算値をペイロード（Ｄ１）のＤ１．Ａに引き当て、乱数$X_n$のハッシュ値をペイロード（Ｄ１）のＤ１．Ｈに引き当てて個別識別子ＥＰＩＤと共に中継ゲートウェイに送信してアテスト要求を行う。ペイロード（Ｄ１）の構成は初期化時の認証の場合と変わらない。

【００５７】

中継ゲートウェイではマスターキーと現在の第２の認証子$C_n$を取得して、図５の説明と同様に演算値１～３を求めて、Ｄ１．Ｈの値と演算値３によりアテストを行う。

アテストが成功すると、中継ゲートウェイは新たな乱数$Y_n$を生成し、乱数$Y_n$と第２の認証子$C_n$とのＸＯＲ演算により新たな第２の認証子$C_{n+1}$を生成する。さらに、新たな第２の認証子$C_{n+1}$をマスターキーでエンコード処理を行い新たな第１の認証子Ａ

$_{n+1}$を生成する。

【００５８】

アプリケーションゲートウェイでは、新たな第２の認証子$C_{n+1}$を受信して、エンドポイント管理テーブルの現在の認証子$C_n$を新たな第２の認証子$C_{n+1}$に更新する。さらに、現在の認証子$C_n$を新たな第２の認証子$C_{n+1}$でエンコード処理を行い、この結果を新たなタイムラインＩＤである$T_n$として個別識別子ＥＰＩＤ、タイムスタンプと共にタイムラインログに記録する。

【００５９】

エンドポイントでは、中継ゲートウェイにて演算値１とのＸＯＲ演算値に変換された新たな第１の認証子$A_{n+1}$を受信して、現在の第１の認証子$A_n$のハッシュ値とのＸＯＲ演算により新たな第１の認証子$A_{n+1}$を取り出し、現在の第１の認証子を更新する。

アテスト時の認証でも、エンドポイントと中継ゲートウェイとの間の通信には、ハッシュ関数によって暗号化された認証子が送受信されるため、元の認証子が露出することはない。

【００６０】

図７は、本発明の実施形態によるタイムライン管理方法を示す図である。

図５や図６で説明したように中継ゲートウェイで認証が成功する毎に、新たな認証子が生成され、それに伴いエンドポイント管理テーブルの認証子が更新され、タイムラインログに記録が追加される。

【００６１】

エンドポイント管理テーブルに保存される第２の認証子は、エンドポイントの個別識別子ＥＰＩＤに関連付けられた第２の認証子の現在値のみであり、更新前の第２の認証子は残らない。これに対しタイムラインログには、初期のタイムラインＩＤ以降の全ての更新情報が残っている。タイムラインＩＤは更新前の第２の認証子を新たな第２の認証子でエンコード処理した値である。そこで例えば新たな第２の認証子を$C_n$とすると更新前の第２の認証子は$C_{n-1}$となるのでこの更新時のタイムラインＩＤである$T_{n-1}$は

$$T_{n-1}=Encode(C_{n-1},C_n)\cdots(3)$$

で表される。

この更新の結果、エンドポイント管理テーブルには、新たな第２の認証子$C_n$が保存され、タイムラインログの最後には更新時のタイムラインＩＤである$T_{n-1}$が記録される。

【００６２】

エンコード処理に使用する関数は、前述の（１）、（２）式で表されるように同じ値を使用してデコード処理を行うと元に戻る性質のある関数である。そこでタイムラインＩＤである$T_{n-1}$をエンドポイント管理テーブルに保存された第２の認証子$C_n$でデコード処理すると更新前の第２の認証子$C_{n-1}$を求めることができる。

同様に一つ前のタイムラインＩＤである$T_{n-2}$を上記の第２の認証子$C_{n-1}$でデコード処理するとさらに一つ前の第２の認証子$C_{n-2}$を求めることができる。こうして順次遡ることで第２の認証子の初期値$C_0$を求めることができる。第２の認証子の初期値$C_0$が求まれば、この値をマスターキーでエンコード処理することにより第１の認証子の初期値$A_0$を求めることができ、必要に応じてエンドポイントの不揮発性メモリに保存された第１の認証子の初期値$A_0$と突き合わせることで、エンドポイントの正当性を確認することができる。

【００６３】

図８は、本発明の実施形態による中継ゲートウェイを経由したエンド・エンドでの共通鍵の配置方法を示す図であり、データ連携の起点と終点、即ちエンドポイントとアプリケーションゲートウェイとの間のデータの送受信に使用される共通の暗号鍵（共通鍵）配置の方法を示す図である。

この段階では、中継ゲートウェイが、エンドポイントで生成した第１の乱数とアプリケーションゲートウェイで生成した第３の乱数を受信し、中継ゲートウェイで生成した第２

の乱数と第１の乱数から暗号鍵の一部を生成してアプリケーションゲートウェイに送信し、第２の乱数と第３の乱数から暗号鍵の他の一部を生成してエンドポイントに送信し、アプリケーションゲートウェイが暗号鍵の一部と第３の乱数から暗合鍵を生成してホワイトリストに保存し、エンドポイントが暗号鍵の他の一部と第１の乱数から暗合鍵を生成して保存する。

【００６４】

前提として、図４及び図５で説明したエンドポイントの初期化及び初期化時の認証が終了しているとする。即ちエンドポイントの現在の第１の認証子は$A_1$であり、エンドポイント管理テーブルに記録された現在の第２の認証子は$C_1$である。

共通鍵の配置段階では、まずエンドポイントにて新たな乱数$x_1$を生成する。共通鍵の配置段階でエンドポイントから送信されるデータのペイロード（Ｄ１）の構成は、識別子ＥＰＩＤと鍵ＫＥＹで構成される。

エンドポイントでは、エンドポイントの個別識別子ＥＰＩＤをペイロードの識別子の値Ｄ１．ＥＰＩＤに引き当て、生成した乱数$x_1$と現在の第１の認証子$A_1$とのＸＯＲ演算結果をペイロードの鍵の値Ｄ１．ＫＥＹに引き当てて中継ゲートウェイに送信する。

【００６５】

中継ゲートウェイでは、個別識別子ＥＰＩＤを取り出してアプリケーションゲートウェイに送信する。

アプリケーションゲートウェイでは、エンドポイント管理テーブルから個別識別子ＥＰＩＤに対応する現在の第２の認証子$C_1$を抽出し、新たに生成した乱数とＸＯＲ演算を行い共通鍵の一部$z_1$を生成する。共通鍵の一部$z_1$は、マスターキー、現在の第２の認証子$C_1$と共に中継ゲートウェイに送信される。

【００６６】

中継ゲートウェイでは、受信した現在の第２の認証子$C_1$をマスターキーでエンコード処理して現在の第２の認証子$C_1$に対応した現在の第１の認証子$A_1$を求める。

続いてエンドポイントから受信したペイロードの鍵の値Ｄ１．ＫＥＹとここで求められた現在の第１の認証子$A_1$とのＸＯＲ演算を行う。Ｄ１．ＫＥＹは乱数$x_1$と現在の第１の認証子$A_1$とのＸＯＲ演算結果であるので、この演算により乱数$x_1$を取り出すことができる。

【００６７】

さらに中継ゲートウェイは新たな乱数$y_1$を生成する。これで暗号鍵の構成要素となる３つの乱数$x_1$、$y_1$、$z_1$がすべてそろうことになるが、中継ゲートウェイでは暗号鍵を完成させることなく、相異なる２つの暗号鍵の一部を生成して一方をアプリケーションゲートウェイに、他方をエンドポイントに送信する。実施形態では、乱数$x_1$と乱数$y_1$のＸＯＲ演算値で生成した暗号鍵の一部をアプリケーションゲートウェイに送信し、乱数$y_1$と乱数$z_1$のＸＯＲ演算値で生成した暗号鍵の一部をエンドポイントに送信する。

【００６８】

アプリケーションゲートウェイでは受信した暗号鍵の一部に乱数$z_1$をＸＯＲ演算することで暗号鍵Ｋ１を生成する。生成した暗号鍵Ｋ１はエンドポイント管理テーブルに個別識別子ＥＰＩＤと関連付けして保存される。

一方エンドポインとでは受信した暗号鍵の一部に乱数$x_1$をＸＯＲ演算することで暗号鍵Ｋ１を生成する。

こうしてアプリケーションゲートウェイとエンドポイントで生成した暗号鍵Ｋ１はともに３つの乱数$x_1$、$y_1$、$z_1$のＸＯＲ演算値となり、暗号に使用される共通鍵をデータ連携の起点と終点で共有することができる。

【００６９】

このようにデータ連携の両端に共通鍵を配置する段階で、通信経路上には完成された共通鍵が一度も露出することがないため、第３者から盗用されることなく安全に共通の暗号鍵の配置を行うことができる。また共通鍵の構成要素となる乱数自体も、認証子によって暗号化されて送信されるため、安全に送信されて盗用を防ぐことができる。

【００７０】
図９は、本発明の実施形態による中継ゲートウェイを経由したエンド・エンドでの暗号通信の鍵の交換方法を示す図である。
図９は図８で説明した共通鍵の配置に続くもので、既に配置された共通鍵を交換して再配置するときの流れを示す。共通鍵の交換は通信セッションごとに行ってもよいし、所定の期間ごとに行ってもよい。
【００７１】
共通鍵の配置が終了した時点では、エンドポイントとアプリケーションゲートウェイはともに共通鍵Ｋ１を保存している。
この随時鍵交換段階においても、エンドポイントから送信されるデータのペイロード（Ｄ１）の構成は、識別子ＥＰＩＤと鍵ＫＥＹで構成される。
まずエンドポイントにて新たな乱数$x_2$を生成し、エンドポイントの個別識別子ＥＰＩＤをペイロードの識別子の値Ｄ１．ＥＰＩＤに引き当て、生成した乱数$x_2$と現在の共通鍵Ｋ１とのＸＯＲ演算結果をペイロードの鍵の値Ｄ１．ＫＥＹに引き当てて中継ゲートウェイに送信する。
【００７２】
中継ゲートウェイは、新たな乱数$y_2$を生成し、乱数$y_2$とエンドポイントから受信したペイロードの鍵の値Ｄ１．ＫＥＹとのＸＯＲ演算を行い、この結果（Ｄ２．ＫＥＹ）を受信した個別識別子ＥＰＩＤと共にアプリケーションゲートウェイに送信する。
【００７３】
アプリケーションゲートウェイは、エンドポイント管理テーブルから個別識別子ＥＰＩＤに関連付けられた共通鍵Ｋ１を抽出し、受信したＤ２．ＫＥＹと共通鍵Ｋ１とのＸＯＲ演算を行い、演算値（Ｔｅｍｐ）を求める。Ｄ２．ＫＥＹは乱数$x_2$とエンドポイントが保存する共通鍵Ｋ１とのＸＯＲ演算結果に乱数$y_2$をさらにＸＯＲ演算した結果であるから、アプリケーションゲートウェイで共通鍵Ｋ１によりＸＯＲ演算を行うことにより、乱数$x_2$と乱数$y_2$とのＸＯＲ演算結果を求めたことになる。
【００７４】
続いてアプリケーションゲートウェイは、新たな乱数$z_2$を生成し、演算値（Ｔｅｍｐ）と乱数$z_2$のＸＯＲ演算を行い、新たな暗号鍵Ｋ２を生成する。結果的に、暗号鍵Ｋ２はエンドポイントで生成した乱数$x_2$、中継ゲートウェイで生成した乱数$y_2$、及びアプリケーションゲートウェイで生成した乱数$z_2$の３つの乱数のＸＯＲ演算値となる。
生成した暗号鍵Ｋ２は現在の共通鍵Ｋ１を使って暗号化、即ち暗号鍵Ｋ２と共通鍵Ｋ１とのＸＯＲ演算を行い、その結果を個別識別子ＥＰＩＤと共に中継ゲートウェイ経由でエンドポイントに送信する。これに伴い、エンドポイント管理テーブルの個別識別子ＥＰＩＤに関連付けられた共通鍵Ｋ１を新たな共通鍵となる暗号鍵Ｋ２に更新して保存する。
【００７５】
エンドポイントでは、受信した暗号鍵Ｋ２と共通鍵Ｋ１とのＸＯＲ演算結果に、エンドポイントが保存する共通鍵Ｋ１でさらにＸＯＲ演算を行い、暗号鍵Ｋ２を取り出す。これによりエンドポイントはアプリケーションゲートウェイと共通の暗号鍵Ｋ２を取得でき、この共通鍵Ｋ２で現在の共通鍵Ｋ１のリプレースを行い保存する。
【００７６】
以上、最初の共通鍵配置後の鍵交換を説明したが、２回目以降の鍵交換についても同様に、アプリケーションゲートウェイがエンドポイント、中継ゲートウェイ、及びアプリケーションゲートウェイのそれぞれで新たに生成した乱数を使用して次の通信用の新たな暗号鍵を生成してエンドポイントに送信し、エンドポイント、アプリケーションゲートウェイのそれぞれが暗号鍵を更新するという方法は変わらない。この際の乱数のやり取りには、現在の共通鍵によるＸＯＲ演算によって暗号化した上で送受信が行われるため、通信経路に暗号鍵の要素となる乱数がそのままの形でやり取りされることはない。したがって安全に共通鍵の交換を行うことができる。
【００７７】

図１０は、本発明の実施形態による中継ゲートウェイを経由したエンド・エンドでの暗号通信の鍵の交換とデータの送受信方法を示す図であり、エンドポイントとアプリケーションゲートウェイの間の鍵の交換を伴う基本的なデータの流れを示す。

図１０は、最初の共通鍵の配置がなされた後のデータの送受信を示し、最後に共有された共通鍵がＫ１であることを示すが、複数回の鍵交換が行われた後、最後に共有された共通鍵がＫｎであったとしても添え字が変わるのみで、基本的なデータの送受信の方法は変わらない。

【００７８】

データの送受信におけるペイロードは、識別子（ＥＰＩＤ）、ファンクションＩＤ（ＦＵＮＣ＿ＩＤ）、鍵（ＫＥＹ）、データ（Ｄａｔａ）の４つの要素で構成される。

データの送信に際し、エンドポイントは新たな乱数$x_2$を生成する。エンドポイントでは、エンドポイントの個別識別子ＥＰＩＤをペイロードの識別子の値Ｄ１．ＥＰＩＤに引き当て、生成した乱数$x_2$と現在の共通鍵Ｋ１とのＸＯＲ演算結果をペイロードの鍵の値Ｄ１．ＫＥＹに引き当て、送信するデータを現在の共通鍵Ｋ１でエンコード処理した結果をペイロードのデータの値Ｄ１．Ｄａｔａに引き当てて中継ゲートウェイに送信する。

【００７９】

中継ゲートウェイは、新たな乱数$y_2$を生成し、乱数$y_2$とエンドポイントから受信したペイロードの鍵の値Ｄ１．ＫＥＹとのＸＯＲ演算を行い、この結果（Ｄ２．ＫＥＹ）を受信した個別識別子ＥＰＩＤ、データと共にアプリケーションゲートウェイに送信する。

【００８０】

アプリケーションゲートウェイは、エンドポイント管理テーブルから個別識別子ＥＰＩＤに関連付けられた共通鍵Ｋ１を抽出し、受信したＤ２．ＫＥＹと共通鍵Ｋ１とのＸＯＲ演算を行い、演算値（Ｔｅｍｐ）を求める。この処理は図９で説明したのと同様、乱数$x_2$と乱数$y_2$とのＸＯＲ演算結果を求めたことになる。

また、アプリケーションゲートウェイは、受信したデータを抽出した共通鍵Ｋ１でデコード処理を行う。エンドポインとから送信されたデータは共通鍵Ｋ１でエンコード処理されているため、前述の（１）、（２）式の関係に基づきデコード処理により、暗号化処理されていないデータが取り出される。

【００８１】

続いてアプリケーションゲートウェイは、新たな乱数ｚ２を生成し、演算値（Ｔｅｍｐ）と乱数$z_2$のＸＯＲ演算を行い、新たな暗号鍵Ｋ２を生成する。新たな暗号鍵Ｋ２はエンドポイントに送信する前に、共通鍵Ｋ１にてＸＯＲ演算にて暗号化される。

また、アプリケーションゲートウェイは、受信したデータへの応答を共通鍵Ｋ１にてエンコード処理してエンドポイントに送信するデータを生成する。

【００８２】

このように共通鍵Ｋ１を使って暗号化処理した暗号鍵Ｋ２及び応答は個別識別子ＥＰＩＤと共に中継ゲートウェイを経由してエンドポイントに送信される。これに伴い、エンドポイント管理テーブルの個別識別子ＥＰＩＤに関連付けられた共通鍵Ｋ１を新たな共通鍵となる暗号鍵Ｋ２に更新して保存する。

【００８３】

エンドポイントでは暗号化された暗号鍵Ｋ２を、共通鍵Ｋ１にてＸＯＲ演算を行って復号し、共通鍵Ｋ２として共通鍵Ｋ１の代わりに置き換えて保存する。さらに受信した応答を共通鍵Ｋ１にてデコード処理を行って復号して取り出す。

このようにすることで、エンドポイントで、データと次の通信用の鍵の要素となる乱数とを共通鍵で暗号化して送信し、アプリケーションゲートウェイで、３つの乱数で生成した次の暗号通信用の共通鍵と、応答とを共通鍵で暗号化して送信することにより、データの送受信と暗号用の共通鍵の交換をまとめて安全に行うことができる。

【００８４】

通信の安全性の面では図１０に示す実施形態の様に、データの送受信毎に暗号鍵を更新することが望ましいが、その分通信に係る負荷も増加するため、前述のように鍵の交換は

必ずしも毎回の通信で行わなくてもよい。通信経路の安全性が確保された前提でデータ送受信のみを行う場合は、各構成要素での乱数の生成は行わず、共通鍵で暗号化したデータを送信し、それに対し共通鍵で暗号化された応答が返信される形でデータの送受信が行われる。

データの送受信と共通鍵の交換は独立して行ってもよく、図１０のようにまとめて行ってもよい。さらには必要に応じこれらを組み合わせてもよい。

【００８５】

以上本発明に実施形態を具体的に説明してきたが、本発明に係るセキュアネットワーク通信方法の特徴を整理すると次のようになる。

１）通信プロトコルに依存しない暗号方式

乱数の発生の予測困難性に依存した方式である。エンドポイントにて送信直前で暗号化され、アプリケーションゲートウェイにてアソシエートアプリケーションの直前で復号されるため、その間の通信路の通信のプロトコルが何層に連携されていても一度も平文に戻ることが無く、通信の方式に依存しない特長がある。

【００８６】

２）簡易な認証と鍵交換のプロトコル（ＳＳＬ等と比較して）

ハッシュ関数と、エンコード・デコード処理に使用する暗号・復号関数によって計算されるエンドポイントとアプリケーションゲートウェイとで非対称な認証子を、予め管理情報に登録することで、それ以外のエンドポイントを「中継ゲートウェイ」で拒絶することができる。また、図８に示すように、２つの鍵長のエレメントの通信とハッシュ関数、そしてＸＯＲ演算という論理演算によって認証から鍵交換のプロトコル間の処理を行うので、ＲＳＡ（公開鍵）方式のような大きな計算を必要としない効率的な方法である。このため、８ＫバイトのＲＡＭ程度の小さなＬＳＩで実装されるエンドポイントでも問題なく動かすことが可能である。また共通鍵の交換のプロトコルの過程では、エンドポイントと中継ゲートウェイ、そしてアプリケーションゲートウェイで発生した３つの乱数から鍵を生成する乱数の発生の予測困難性を根拠とした共通鍵暗号方式であり、それぞれの乱数がお互いに独立という前提で暗号化の強度が決まる特徴がある。乱数の発生は独立という特徴のため、半導体の熱雑音をエントロピーソースとするような自然現象を利用した乱数発生方式を利用することができ、共通鍵の予測困難性、つまり強度を高めることができる。

【００８７】

３）ホワイトリストとタイムラインログによる認証となりすましの予防

管理情報として記録されるログは、単なる時間由来のタイムスタンプではなく、認証子を暗号化して計算されるタイムラインＩＤを持つ。このタイムラインＩＤは乱数から計算された認証子と１つ前のタイムラインＩＤを鍵にして暗号化して計算されるため、乱数の発生確率に基づく精度を持ち、不可逆性が維持されるという特性を持つ。また、このタイムラインＩＤは、管理情報に持つ認証子と最後に記録されたタイムランＩＤにより、１世代前の認証子が計算できる。この性質を利用して、タイムラインログを順次遡ることによってエンドポイントの不揮発性メモリに記録された認証子の初期値を計算することができる。なりすましが発生しても、管理情報の認証子とタイムラインログにより、ユニーク性を持つエンドポイントを特定することができる。

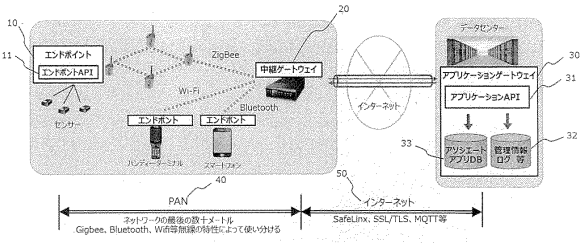
【００８８】

４）冗長性の少ないストリーム暗号であり、計算に頼らない効率的な暗号／復号方式

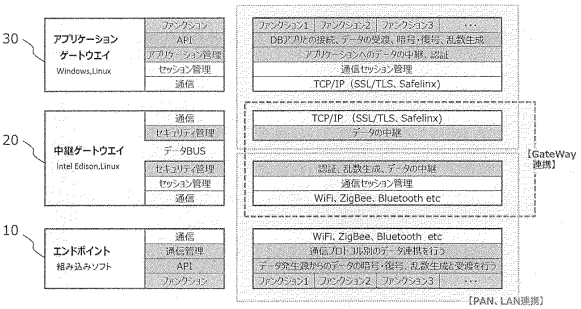
共通鍵の交換の段階ではＸＯＲの論理演算、ハッシュ関数のみが使われる。通信のオーバーヘッドは鍵長の二倍程度でありＳＳＬのような鍵交換プロトコルで行われる計算量及び通信量と比較してはるかに効率的である。また、暗号・復号の過程でも直前の共通鍵で初期化された疑似乱数生成器から得られる乱数ストリームと、送信電文のデータストリームと同じオフセットのビットをＸＯＲで暗号化する方式であるために、計算量が少ないというメリットの他に、暗号化の計算過程で計算用のバッファなどを必要としない特長がある。このような方式のために他の暗号方式に比較しても小さなＬＳＩでも実装が可能となる。

【００８９】
５）シームレス
データを中継する伝送路での暗号を必要としない。つまり、エンドポイントのアプリケーション層で暗号化して、データを処理するアソシエートアプリケーションの層で復号化されるため、その間で一度も平文にならない。また、通信のプロトコル等にも依存しないエンド・エンドでのシームレスなデータ連携が可能となる。
【００９０】
６）ファンクション単位の独立性
通信の経路の前と後に暗号・復号の出入り口があり、ＡＰＩはさらに論理的な機能を独立に設定して、システムの拡張性を考慮できるように実装することができる。アプリケーションＡＰＩはファンクションのＩＤ単位に動的に接続するような仕組みにすること、ファンクション単位にアソシエートアプリケーションを維持管理できるような仕組みにすることによって、ファンクション単位に運用の独立性を持たせることができる。
【００９１】
以上、本発明の実施形態について図面を参照しながら詳細に説明したが、本発明は、上述の実施形態に限定されるものではなく、本発明の技術的範囲から逸脱しない範囲内で多様に変更することが可能である。
【符号の説明】
【００９２】
１０　エンドポイント
１１　エンドポイントＡＰＩ
２０　中継ゲートウェイ
３０　アプリケーションゲートウェイ
３１　アプリケーションＡＰＩ
３２　アソシエートアプリＤＢ
３３　ネットワーク情報管理メモリ
４０　ＰＡＮ
５０　インターネット

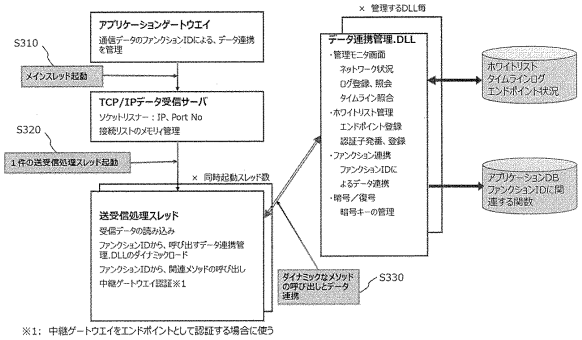
【図１】



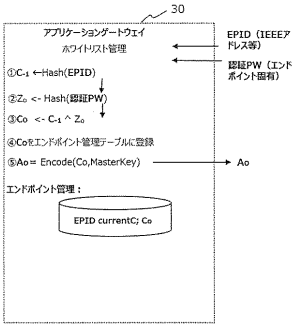

【図２】



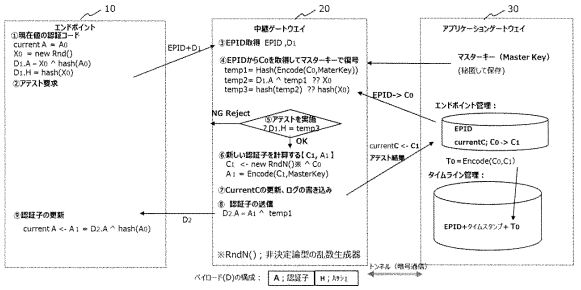

【図３】



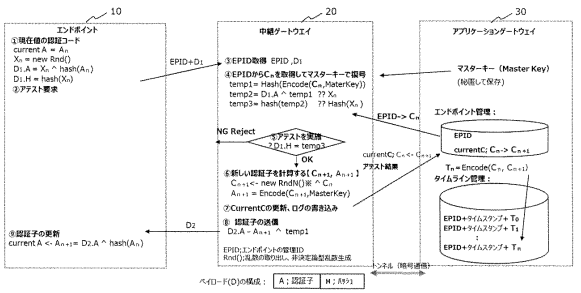

【図４】



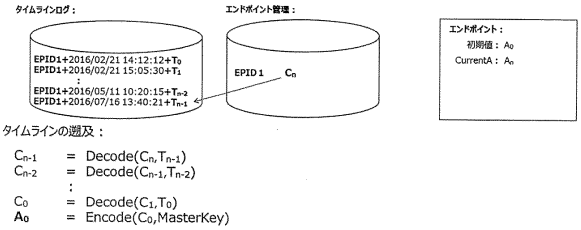

【図５】




【図６】



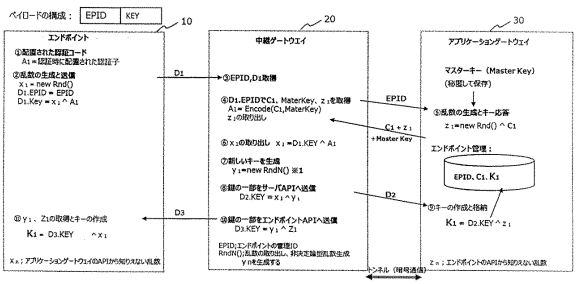

【図７】



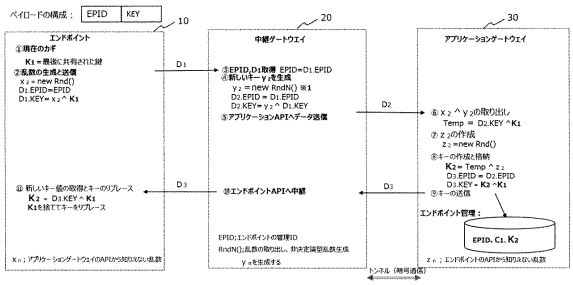

タイムラインの遡及：

$C_{n-1} = Decode(C_n, T_{n-1})$
$C_{n-2} = Decode(C_{n-1}, T_{n-2})$
：
$C_0 = Decode(C_1, T_0)$
$A_0 = Encode(C_0, MasterKey)$

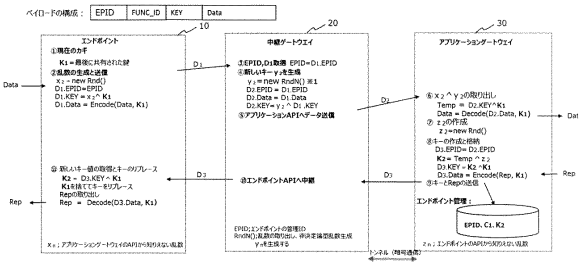
【図８】



【図９】


ペイロードの構成：
EPID
KEY
10
20
30
エンドポイント
①現在のカギ
K1＝最後に共有された鍵
②乱数の生成と送信
x2 = new Rnd()
D1.EPID=EPID
D1.KEY= x2 ^ K1
中継ゲートウエイ
③EPID,D1取得 EPID=D1.EPID
④新しいキーy2を生成
y2 =new RndN() ※1
D2.EPID = D1.EPID
D2.KEY= y2 ^ D1.KEY
⑤アプリケーションAPIへデータ送信
アプリケーションゲートウエイ
⑥ x2 ^ y2の取り出し
Temp = D2.KEY ^K1
⑦ z2の作成
z2 =new Rnd()
⑧キーの作成と格納
K2= Temp ^ z2
D3.EPID = D2.EPID
D3.KEY = K2 ^K1
⑨キーの送信
⑩ 新しいキー値の取得とキーのリプレース
K2 = D3.KEY ^ K1
K1を捨ててキーをリプレース
⑪エンドポイントAPIへ中継
エンドポイント管理：
EPID、C1、K2
D1
D2
D3
EPID:エンドポイントの管理ID
RndN():乱数の取り出し。非決定論型乱数生成
y2を生成する
xn：アプリケーションゲートウェイのAPIから知られない乱数
zn：エンドポイントのAPIから知られない乱数
トンネル（暗号通信）

【図１０】


ペイロードの構成：
EPID
FUNC_ID
KEY
Data
10
20
30
エンドポイント
①現在のカギ
K1＝最後に共有された鍵
②乱数の生成と送信
x2 = new Rnd()
D1.EPID=EPID
D1.KEY = x2^ K1
D1.Data = Encode(Data, K1)
Data
中継ゲートウエイ
③EPID,D1取得 EPID=D1.EPID
④新しいキーy2を生成
y2 = new RndN() ※1
D2.EPID = D1.EPID
D2.Data = D1.Data
D2.KEY= y2 ^ D1 .KEY
⑤アプリケーションAPIへデータ送信
アプリケーションゲートウエイ
⑥ x2 ^ y2の取り出し
Temp = D2.KEY^K1
Data = Decode(D2.Data, K1)
Dat
⑦ z2の作成
z2=new Rnd()
⑧キーの作成と格納
D3.EPID= D2.EPID
K2= Temp ^ z2
D3.KEY = K2 ^K1
D3.Data = Encode(Rep, K1)
Rep
⑨キーとRepの送信
⑩ 新しいキー値の取得とキーのリプレース
K2 = D3.KEY ^ K1
K1を捨ててキーをリプレース
Repの取り出し
Rep = Decode(D3.Data, K1)
Rep
⑪エンドポイントAPIへ中継
エンドポイント管理：
EPID、C1、K2
D1
D2
D3
EPID:エンドポイントの管理ID
RndN():乱数の取り出し。非決定論型乱数生成
y2を生成する
xn：アプリケーションゲートウェイのAPIから知られない乱数
zn：エンドポイントのAPIから知られない乱数
トンネル（暗号通信）

フロントページの続き

(56)参考文献 特開２０１５－１５８６１０（ＪＰ，Ａ）
特開２０１６－１２８９９８（ＪＰ，Ａ）
特開平１１－５５２４８（ＪＰ，Ａ）
特表２０１３－５１６１４９（ＪＰ，Ａ）
国際公開第２０１６／１４７３８２（WO，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　　９／０８
Ｇ０６Ｆ　２１／４４