

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7607599号
(P7607599)

(45)発行日 令和6年12月27日(2024.12.27)

(24)登録日 令和6年12月19日(2024.12.19)

(51)国際特許分類

FI

H 0 4 L 9/08 (2006.01) H 0 4 L 9/08 C

H 0 4 L 9/32 (2006.01) H 0 4 L 9/32 2 0 0 A

請求項の数 28 (全39頁)

(21)出願番号	特願2021-575035(P2021-575035)	(73)特許権者	590000248
(86)(22)出願日	令和2年6月11日(2020.6.11)		コーニンクレッカ フィリップス エヌ
(65)公表番号	特表2022-537733(P2022-537733 A)		ヴェ
(43)公表日	令和4年8月29日(2022.8.29)		Koninklijke Philips N. V.
(86)国際出願番号	PCT/EP2020/066179		オランダ国 5 6 5 6 アーヘー アイン
(87)国際公開番号	WO2020/254177		ドーフエン ハイテック キャンパス 5 2
(87)国際公開日	令和2年12月24日(2020.12.24)		High Tech Campus 52,
審査請求日	令和5年6月8日(2023.6.8)		5 6 5 6 AG Eindhoven, N
(31)優先権主張番号	19181035.7		etherlands
(32)優先日	令和1年6月18日(2019.6.18)	(74)代理人	110001690
(33)優先権主張国・地域又は機関	欧州特許庁(EP)		弁理士法人M&Sパートナーズ
		(72)発明者	ガルシア モーション オスカー
			オランダ国 5 6 5 6 アーヘー アイン
			ドーフエン ハイ テック キャンパス 5
			最終頁に続く

(54)【発明の名称】 認証付き鍵共有

(57)【特許請求の範囲】

【請求項1】

第1の暗号デバイスを用いて格子ベース鍵交換または格子ベース鍵カプセル化の鍵共有
プロトコルを実行するための第2の暗号デバイスであって、前記第2の暗号デバイスは、
前記格子ベース鍵交換または前記格子ベース鍵カプセル化の鍵共有プロトコルを使用し
て前記第1の暗号デバイスと通信する通信インターフェースと、

プロセッサとを備え、前記プロセッサは、

前記第1の暗号デバイスから受け取られたプレシード、前記第2の暗号デバイスと前
記第1の暗号デバイスとの間で事前に共有される事前共有秘密、およびさらなる情報から
最終シードを計算し、

決定論的乱数生成器または疑似乱数生成器を使用して前記最終シードから共通オブジ
ェクトを計算し、

前記第1の暗号デバイスに関連付けられた第1の公開鍵を取得し、前記第2の暗号デ
バイスに関連付けられた第2の秘密鍵を取得し、ここで、前記第1の暗号デバイスが決定
論的乱数生成器または疑似乱数生成器を使用して前記最終シードから前記共通オブジェ
クトを計算し、前記第1の公開鍵は前記第1の暗号デバイスにおいて計算され、当該計算は
、前記第1の暗号デバイスのプロセッサが計算した前記共通オブジェクトと前記第1の暗
号デバイスの第1の秘密鍵との間の乗算を含み、

前記第2の秘密鍵から第2の公開鍵を計算し、ここで、前記第2の公開鍵を計算する
ことは前記第2の秘密鍵を前記第2の暗号デバイスのプロセッサが計算した前記共通オブ

ジェクトと乗算することを含み、

前記第1の公開鍵および前記第2の秘密鍵から第2の生の共有鍵を計算し、ここで、前記第2の生の共有鍵の計算は、前記第2の秘密鍵と前記第1の公開鍵との間の乗算を含み、

前記第2の公開鍵を前記第1の暗号デバイスに伝送し、

前記さらなる情報は、

前記第1の暗号デバイスのネットワークアドレス、

前記第2の暗号デバイスのネットワークアドレス、

ネットワーク接続の目的、

時刻、および、

前記第1の暗号デバイスと前記第2の暗号デバイスによって維持され、少なくとも共有鍵が導出されるたびに増加されるインタラクションカウンタの少なくとも1つを含む、第2の暗号デバイス。

【請求項2】

前記事前共有秘密は事前共有パスワードを含む、請求項1に記載の第2の暗号デバイス。

【請求項3】

前記プレシードは平文形式で送受信される、請求項1または2に記載の第2の暗号デバイス。

【請求項4】

前記プレシードは前記通信インターフェースを介して送受信されるが、前記事前共有秘密は前記通信インターフェースを介して送受信されない、請求項1から3のいずれか一項に記載の第2の暗号デバイス。

【請求項5】

前記最終シードはさらに、前記第2の暗号デバイスと前記第1の暗号デバイスとの間のネットワーク接続からのネットワーク構成データから計算される、請求項1から4のいずれか一項に記載の第2の暗号デバイス。

【請求項6】

前記第2の暗号デバイスの前記プロセッサは、

送信鍵を生成し、

カプセル化関数を適用することによって、前記第2の生の共有鍵の少なくとも一部を用いて前記送信鍵をカプセル化して、カプセル化されたデータを取得し、

前記カプセル化されたデータを前記第1の暗号デバイスに送る、請求項1から5のいずれか一項に記載の第2の暗号デバイス。

【請求項7】

前記第2の暗号デバイスの前記プロセッサは、

前記生の共有鍵の係数の少なくとも一部のための調整データを生成し、ここで、前記調整データは、前記第1および第2の暗号デバイスにおいて導出された前記第1および第2の生の共有鍵の間の差を低減することを可能にする情報を含み、

前記第2の生の共有鍵から送信鍵を生成し、

前記調整データを前記第1の暗号デバイスに送る、請求項1から6のいずれか一項に記載の第2の暗号デバイス。

【請求項8】

前記第2の公開鍵を計算する際の前記共通オブジェクトとの前記乗算はノイズな乗算である、請求項1から7のいずれか一項に記載の第2の暗号デバイス。

【請求項9】

前記第2の暗号デバイスの前記プロセッサは、前記送信鍵を用いてメッセージを暗号化して、前記暗号化されたメッセージを前記第1の暗号デバイスに送り、かつ／または

前記第2の暗号デバイスの前記プロセッサは、前記送信鍵を用いて前記暗号化されたメッセージを復号して、前記メッセージを検証する、請求項6または7に記載の第2の暗号デバイス。

10

20

30

40

50

【請求項 10】

前記送信鍵はランダムであり、かつ／または一時的であり、かつ／または対称であり、かつ／または前記第1の公開鍵から独立している、請求項6または7に記載の第2の暗号デバイス。

【請求項 11】

前記第2の公開鍵、前記第2の秘密鍵、前記第2の生の共有鍵、および前記共通オブジェクトは有限体または環上の行列であるか、又は、

前記第2の公開鍵、前記第2の秘密鍵、前記第2の生の共有鍵、および前記共通オブジェクトは有限体または環上の多項式である、請求項1から10のいずれか一項に記載の第2の暗号デバイス。

【請求項 12】

ネットワーク構成データは、前記送信鍵を用いて暗号化されて送られる、請求項6または7に記載の第2の暗号デバイス。

【請求項 13】

第2の暗号デバイスを用いて格子ベース鍵交換または格子ベース鍵カプセル化の鍵共有プロトコルを実行するための第1の暗号デバイスであって、前記第1の暗号デバイスは、

前記格子ベース鍵交換または前記格子ベース鍵カプセル化の鍵共有プロトコルを使用して前記第2の暗号デバイスと通信する通信インターフェースと、

プロセッサとを備え、前記プロセッサは、

プレシードを選択し、前記プレシードを前記第2の暗号デバイスに送り、

前記プレシード、前記第2の暗号デバイスと前記第1の暗号デバイスとの間で事前に共有される事前共有秘密、およびさらなる情報から最終シードを計算し、

決定論的乱数生成器または疑似乱数生成器を使用して前記最終シードから共通オブジェクトを計算し、

前記第1の暗号デバイスに関連付けられた第1の秘密鍵を取得し、前記第1の秘密鍵を前記共通オブジェクトと乗算することを含む計算により、前記第1の秘密鍵から第1の公開鍵を計算し、前記第1の公開鍵を前記第2の暗号デバイスに伝送し、

前記第2の暗号デバイスから第2の公開鍵を受け取り、

前記第2の公開鍵および前記第1の秘密鍵から第1の生の共有鍵を計算し、ここで、前記第1の生の共有鍵の計算は、前記第2の公開鍵と前記第1の秘密鍵との間の乗算を含み、

前記さらなる情報は、

前記第1の暗号デバイスのネットワークアドレス、

前記第2の暗号デバイスのネットワークアドレス、

ネットワーク接続の目的、

時刻、および、

前記第1の暗号デバイスと前記第2の暗号デバイスによって維持され、少なくとも共有鍵が導出されるたびに増加されるインタラクションカウンタの少なくとも1つを含む、第1の暗号デバイス。

【請求項 14】

前記事前共有秘密は事前共有パスワードを含む、請求項13に記載の第1の暗号デバイス。

【請求項 15】

前記プレシードは平文形式で送受信される、請求項13または14に記載の第1の暗号デバイス。

【請求項 16】

前記プレシードは前記通信インターフェースを介して送受信されるが、前記事前共有秘密は前記通信インターフェースを介して送受信されない、請求項13から15のいずれか一項に記載の第1の暗号デバイス。

【請求項 17】

前記最終シードはさらに、前記第2の暗号デバイスと前記第1の暗号デバイスとの間のネットワーク接続からのネットワーク構成データから計算される、請求項13から16のいずれか一項に記載の第1の暗号デバイス。

【請求項18】

前記第1の暗号デバイスの前記プロセッサは、

前記第2の暗号デバイスからカプセル化されたデータを受け取り、

前記第1の生の共有鍵の少なくとも一部を使用して、前記カプセル化されたデータのカプセル化を解除することで送信鍵を取得する、請求項13から17のいずれか一項に記載の第1の暗号デバイス。

【請求項19】

前記第1の暗号デバイスの前記プロセッサは、

前記第2の暗号デバイスから調整データを受け取り、

前記第1の生の共有鍵内の係数の少なくとも一部に対して、調整関数における前記調整データを適用し、

前記調整データを適用された第1の生の共有鍵から送信鍵を生成する、請求項13から18のいずれか一項に記載の第1の暗号デバイス。

【請求項20】

前記第1の公開鍵を計算する際の前記共通オブジェクトとの前記乗算はノイズな乗算である、請求項13から19のいずれか一項に記載の第1の暗号デバイス。

【請求項21】

前記第1の暗号デバイスの前記プロセッサは、前記第2の暗号デバイスから暗号化されたメッセージを受け取り、前記送信鍵を用いて前記暗号化されたメッセージを復号して、前記メッセージを検証し、かつ／または

前記第1の暗号デバイスの前記プロセッサは、前記送信鍵を使用してメッセージを暗号化して、前記暗号化されたメッセージを前記第2の暗号デバイスに送る、請求項18又は19に記載の第1の暗号デバイス。

【請求項22】

前記送信鍵はランダムであり、かつ／または一時的であり、かつ／または対称であり、かつ／または前記第1の公開鍵から独立している、請求項18又は19に記載の第1の暗号デバイス。

【請求項23】

前記第1の公開鍵、前記第1の秘密鍵、前記第1の生の共有鍵、および前記共通オブジェクトは有限体または環上の行列であるか、又は、

前記第1の公開鍵、前記第1の秘密鍵、前記第1の生の共有鍵、および前記共通オブジェクトは有限体または環上の多項式である、請求項13から22のいずれか一項に記載の第1の暗号デバイス。

【請求項24】

ネットワーク構成データは、前記送信鍵を用いて暗号化されて送られる、請求項18又は19に記載の第1の暗号デバイス。

【請求項25】

第1の暗号デバイスを用いて格子ベース鍵交換または格子ベース鍵カプセル化の鍵共有プロトコルを実行するための暗号方法であって、前記暗号方法は、

前記第1の暗号デバイスから受け取られたプレシード、第2の暗号デバイスと前記第1の暗号デバイスとの間で事前に共有される事前共有秘密、およびさらなる情報から最終シードを計算するステップと、

決定論的乱数生成器または疑似乱数生成器を使用して前記最終シードから共通オブジェクトを計算するステップと、

前記第1の暗号デバイスに関連付けられた第1の公開鍵、および前記第2の暗号デバイスに関連付けられた第2の秘密鍵を取得するステップであって、前記第1の暗号デバイスが決定論的乱数生成器または疑似乱数生成器を使用して前記最終シードから前記共通オブ

10

20

30

40

50

ジェクトを計算し、前記第 1 の公開鍵は前記第 1 の暗号デバイスにおいて計算され、当該計算は、前記第 1 の暗号デバイスのプロセッサが計算した前記共通オブジェクトと前記第 1 の暗号デバイスの第 1 の秘密鍵との間の乗算を含む、ステップと、

前記第 2 の秘密鍵から第 2 の公開鍵を計算するステップであって、前記第 2 の公開鍵を計算することは、前記第 2 の秘密鍵に前記第 2 の暗号デバイスのプロセッサが計算した前記共通オブジェクトを乗算することを含む、ステップと、

前記第 1 の公開鍵および前記第 2 の秘密鍵から第 2 の生の共有鍵を計算するステップであって、前記第 2 の生の共有鍵を計算することは、前記第 2 の秘密鍵と前記第 1 の公開鍵との間の乗算を含む、ステップと、

前記第 2 の公開鍵を前記第 1 の暗号デバイスに伝送するステップとを含み、

前記さらなる情報は、

前記第 1 の暗号デバイスのネットワークアドレス、

前記第 2 の暗号デバイスのネットワークアドレス、

ネットワーク接続の目的、

時刻、および、

前記第 1 の暗号デバイスと前記第 2 の暗号デバイスによって維持され、少なくとも共有鍵が導出されるたびに増加されるインタラクションカウンタの少なくとも 1 つを含む、暗号方法。

【請求項 26】

第 2 の暗号デバイスを用いて格子ベース鍵交換または格子ベース鍵カプセル化の鍵共有プロトコルを実行するための暗号方法であって、前記暗号方法は、

プレシードを選択し、前記プレシードを前記第 2 の暗号デバイスに送るステップと、

前記プレシード、前記第 2 の暗号デバイスと第 1 の暗号デバイスとの間で事前に共有される事前共有秘密、およびさらなる情報から最終シードを計算するステップと、

決定論的乱数生成器または疑似乱数生成器を使用して前記最終シードから共通オブジェクトを計算するステップと、

前記第 1 の暗号デバイスに関連付けられた第 1 の秘密鍵を取得し、前記第 1 の秘密鍵を前記共通オブジェクトと乗算することを含む計算により、前記第 1 の秘密鍵から第 1 の公開鍵を計算し、前記第 1 の公開鍵を前記第 2 の暗号デバイスに伝送するステップと、

前記第 2 の暗号デバイスから第 2 の公開鍵を受け取るステップと、

前記第 2 の公開鍵および前記第 1 の秘密鍵から第 1 の生の共有鍵を計算するステップであって、前記第 1 の生の共有鍵を計算することは、前記第 2 の公開鍵と前記第 1 の秘密鍵との間の乗算を含む、ステップとを含み、

前記さらなる情報は、

前記第 1 の暗号デバイスのネットワークアドレス、

前記第 2 の暗号デバイスのネットワークアドレス、

ネットワーク接続の目的、

時刻、および、

前記第 1 の暗号デバイスと前記第 2 の暗号デバイスによって維持され、少なくとも共有鍵が導出されるたびに増加されるインタラクションカウンタの少なくとも 1 つを含む、暗号方法。

【請求項 27】

請求項 25 に記載の暗号方法をプロセッサシステムに実行させる命令を表すデータを含む、非一時的なコンピュータ可読媒体。

【請求項 28】

請求項 26 に記載の暗号方法をプロセッサシステムに実行させる命令を表すデータを含む、非一時的なコンピュータ可読媒体。

【発明の詳細な説明】

【技術分野】

【0001】

10

20

30

40

50

本開示の主題は、第1の暗号デバイス、第2の暗号デバイス、第1の暗号方法、第2の暗号方法、およびコンピュータ可読媒体に関する。

【背景技術】

【0002】

暗号技術において、鍵共有プロトコルは、2つ以上のパーティが鍵を共有することを可能にするプロトコルである。例えば、共有された鍵は、パーティ間のさらなる通信を保護するために、例えば、さらなる通信を認証および／または暗号化するために使用され得る。両パーティ間のすべての通信を盗聴する攻撃者は鍵について何も学んではならない。同じ通信を見る攻撃者は何も学ばないか、またはほとんど学ばないにも関わらず、両パーティ自身は共有鍵を導出することができる。

10

【0003】

Whitfield DiffieとMartin Hellmanが公開鍵暗号法概念を導入した1976年に実用的な鍵共有プロトコルが導入された。彼らは、 q の要素を有する有限体 $GF(q)$ 上で対数を計算することの明らかな困難さを利用した、2者間の鍵共有のためのシステムを提案した。このシステムを使用することで2人のユーザが対称鍵を共有することができる。

【0004】

鍵共有プロトコルは、複数のパーティが通信する多くの用途、例えばIoT (Internet of Things) やアドホックワイヤレスネットワークなどの分野で有用である。鍵共有はデバイス間のリンクを保護するために使用され得る。別の例はリーダーと電子タグ、例えばカードリーダーとスマートカード、またはタグリーダーとタグ、例えばRFIDタグまたはNFCタグの間の通信である。

20

【0005】

パーティ間の安全な通信を容易にするために、鍵共有プロトコルは、時にはさらに暗号鍵交換 (KEX) スキームと暗号鍵カプセル化 (KEM) スキームに細分化される。暗号鍵カプセル化 (KEM) スキームは非対称暗号を使用するスキームであり、公に知られている値 (例えば、公開鍵) および各パーティが持つ秘密の値 (例えば、秘密鍵) を利用して2つのパーティの間で共有される秘密を確立する。

【0006】

KEXスキームは各パーティによる公開鍵の交換を含み、その後、公開鍵は、共通の共有秘密を計算するために、他方のパーティによって各自の秘密鍵と共に独立して使用される。KEXスキームのよく知られた例は上述のDiffie-Hellman鍵交換であり、この方式のセキュリティは離散対数問題を解くことに基づいている。典型的には、どちらのパーティも鍵の選択を強制することができないように、両パーティが結果に影響を及ぼすことができる。いくつかのKEXスキームの興味深い特徴は、実際の最終的な共有される秘密が (暗号化された形式でさえも) 決してパーティ間で交換されず、両者によってそれぞれ独自に計算されることである。これは前方秘匿性として知られる望ましい特徴をもたらし、将来の攻撃者へのパーティの長期秘密鍵の漏洩さえも、過去に交換された暗号化されたメッセージの秘密の漏洩をもたらさないことを保証する。

30

【0007】

KEMスキームは2つのエンティティまたはパーティ間で共有秘密を確立するために、一方のパーティ、通常は通信のイニシエータによる非対称暗号を使用して、レスポンドとして知られる他方のパーティの公開鍵を使用して共有秘密を暗号化し、他方のパーティに送信する。レスポンドはその後、自分の秘密鍵を使用してそれを復号し、イニシエータパーティと安全に通信するためにそれを使用することができる。過去のセッションのための一方のパーティの秘密鍵を不正入手し、そのセッションにおいてパーティ間で交換されたすべてのメッセージを記録した攻撃者はそのセッションの共有秘密を復元できるので、KEMスキームは前方秘匿性を達成できない。

40

【0008】

DiffieおよびHellmanの論文以来、新たな懸念が提起されてきた。IoT

50

のセキュリティニーズが高まっているため、鍵交換スキームはさらに、高効率の要件、および低通信または帯域幅の要件を達成する必要がある。また、鍵共有プロトコルはクラシカルな敵および量子に対応した敵に対して安全であることが好ましい。

【0009】

さらに、これらの鍵共有プロトコルの欠点はパーティを認証しないことである。これにより、中間者（MITM）攻撃に対して脆弱になる。中間者攻撃では、攻撃者は第1のパーティと第2のパーティとの間に身を置き得る。そして、攻撃者は第1の共有鍵を第1のパーティとネゴシエートし、第2の共有鍵を第2のパーティとネゴシエートし得る。したがって、攻撃者はすべてを復号できる。従来、KEX型プロトコルおよびKEM型プロトコルはともに、この問題を回避するために、例えばデジタル署名を必要とする。署名は通常、軽量プロトコルでも回避される動作である。

10

【0010】

これに対処するためのアプローチは、いわゆる認証付き鍵交換（AKE）である。AKEプロトコルは鍵をネゴシエートすると同時に、通信するパーティの身元を認証できる。特に重要なタイプのAKEはPAKE（password-based AKE）である。PAKEは、対称セッション鍵を認証およびネゴシエートするには暗号的に安全でない可能性のある、人間が記憶可能なパスワード（またはパスフレーズ）を利用する。PAKEではパスワードは両パーティによって事前に共有され得る。

【0011】

参照によって本明細書に援用される、Xinwei Gaoらによる論文“Efficient Implementation of Password-Based Authenticated Key Exchange from RLWE and Post-Quantum TLS”では、論文中の図2において、RLWE-PPKと呼ばれるポスト量子パスワードベース認証付き鍵交換（PAKE）プロトコルが提案されている。

20

【0012】

RLWE-PPKは、事前共有パスワードpwdを有する2者間（パーティiおよびパーティj）のプロトコルである。各パーティが共有パスワードから2つの異なるハッシュ $\gamma_1 = H_1(pwd)$ および $\gamma_2 = H_2(pwd)$ を計算し、ハッシュは、他方のパーティに送信する前後に各々のパーティの公開鍵をマスクおよびアンマスクするために使用される。共有鍵を計算するには、各パーティがさらなるハッシュを計算する必要があり、ハッシュの総数はパーティごとに3つとなる。RLWE-PPKは2パス暗黙認証付き鍵プロトコル（two-pass implicitly authenticated key protocol）である。暗黙認証とは、認証の明示的な確認は計算できないが、認証が失敗した場合、両パーティは同じ鍵を取得しないことから、認証されていないパーティに通信が開示されることを防ぐことを意味する。

30

【0013】

この既知のプロトコルにはいくつかの欠点がある。例えば、鍵共有ごとに3つのハッシュを計算する必要がある。リソース制約デバイスの場合これは負担となる。さらに、この既知のプロトコルでは、公開鍵の知識と盗聴された通信からパスワードハッシュを計算できるため、公開鍵を秘密にしておく必要がある。

40

【発明の概要】

【0014】

改善された認証付き鍵共有を備えたデバイスおよび／または方法を有することは有益であろう。本明細書および特許請求の範囲において、上記および他の問題に対処することを目的とした暗号デバイスおよび方法が記載されており、クレームされている。

【0015】

既存の鍵共有プロトコルは、たとえ量子セキュアであったとしても、通常は暗黙的にも明示的にも認証付きではない。既存の鍵共有プロトコルを認証付き鍵共有プロトコルに拡張できれば有益であろう。認証付き鍵共有プロトコルは、2つのパーティが通信チャネルを検証する必要がある現実のシナリオでの使用により適している。

50

【0016】

本開示の主題は、第1の暗号デバイス、第2の暗号デバイス、第1の暗号方法、第2の暗号方法、およびコンピュータ可読媒体を含む。第1および第2の暗号デバイスは、事前共有秘密、例えばパスワードから、および両デバイス間で交換されるプレシードから最終シードを計算するように構成され得る。最終シードは共通オブジェクトを計算するために使用されてもよく、共通オブジェクトは、例えば鍵共有プロトコルを実行するために、公開鍵を計算するために使用されてもよい。

【0017】

実施形態は中間者攻撃を防ぐのに有用である。第1の暗号デバイス（第1のパーティとも呼ばれる）および第2の暗号デバイス（第2のパーティとも呼ばれる）は事前共有秘密を有するため、両デバイスは、公開鍵を導出するために使用される共通オブジェクトを導出することができる。攻撃者はパスワードへのアクセスを有さないので共通オブジェクトを計算できない。したがって、正しい計算を実行するには共通オブジェクトが必要であることから、攻撃者は両パーティ間の通信に割り込むことができない。興味深いことに、パスワードの強度は主に認証の強度に関連し、導出される共有鍵の強度は影響を受けない可能性がある。特に、一実施形態に係る鍵共有プロトコルは認証なしと同程度に強力（例えば、量子耐性）であり得る。

【0018】

実施形態はさらに効率的に実施され得る。例えば、一実施形態では、両パーティは、例えば最終シードを計算するために、単一のハッシュ関数の計算しか行わない可能性がある。必要に応じて、非対称デジタル署名なしで明示認証が得られ得る。また、公開鍵が秘密である必要はない。なお、一実施形態によれば、全てまたはほとんどのNISTポスト量子暗号（PQC）格子ベース候補が認証に適合させられ得る。

【0019】

一実施形態では、プレシードは通信チャネルを介して平文で通信され、一方、事前共有秘密はチャネルを介して通信されないか、または平文では通信されない。例えば、通信チャネルは、鍵共有プロトコルにおいてデバイス間で交換される他のデータのためにも使用され得る。

【0020】

本開示の主題の側面は暗号方法を含む。

【0021】

方法の実施形態は、コンピュータ実装方法としてコンピュータ上に、専用ハードウェアに、またはこれらの組み合わせとして実装されてもよい。方法の実施形態のための実行可能コードはコンピュータプログラム製品上に保存されてもよい。コンピュータプログラム製品の例は、メモリデバイス、光記憶デバイス、集積回路、サーバ、オンラインソフトウェアなどを含む。好ましくは、コンピュータプログラム製品は、プログラム製品がコンピュータ上で実行されるとき、方法の実施形態を実行するためのコンピュータ可読媒体上に保存された非一時的プログラムコードを含む。

【0022】

一実施形態では、コンピュータプログラムは、該コンピュータプログラムがコンピュータ上で実行されたとき、方法の実施形態のステップの全てまたは一部を実行するように構成されたコンピュータプログラムコードを含む。好ましくは、コンピュータプログラムはコンピュータ可読媒体上に具現化される。

【0023】

本開示の主題の他の側面は、コンピュータプログラムをダウンロード可能にする方法を提供する。この態様は、コンピュータプログラムが例えばアップルのApp Store、GoogleのPlay Store、またはMicrosoftのWindows Storeなどにアップロードされており、そのようなストアからコンピュータプログラムをダウンロード可能な場合に使用される。

【図面の簡単な説明】

【0024】

以下、例としてさらなる詳細、側面、および実施形態について図面を参照しながら説明する。図中の要素は簡潔さおよび明瞭さを意図して描かれており、必ずしも縮尺通りに描かれていない。図面において、すでに説明した要素に対応する要素には同じ参照番号が与えられている可能性がある。

【0025】

【図1】図1はKEXプロトコルの実施形態の例を概略的に示す。

【図2】図2はKEMプロトコルの実施形態の例を概略的に示す。

【図3】図3はプロトコルの実施形態の例を概略的に示す。

【図4】図4は、一実施形態に係る暗号システムの例を概略的に示す。

10

【図5a】図5aは、一実施形態に係る暗号方法の例を概略的に示す。

【図5b】図5bは、一実施形態に係る暗号方法の例を概略的に示す。

【図6a】図6aは、一実施形態に係るコンピュータプログラムを含む書き込み可能部分を有するコンピュータ可読媒体を概略的に示す。

【図6b】図6bは、一実施形態に係るプロセッサシステムの表現を概略的に示す。

【符号の説明】

【0026】

10 第1の暗号デバイス

20 第2の暗号デバイス

10.1～10.3 暗号動作

20

11.1 暗号メッセージ

20.1～20.2 暗号動作

21.1～21.2 暗号メッセージ

13 アウトオブバウンド通信

14～15 鍵共有プロトコル

16～17 メッセージ

31 登録フェーズ

32 鍵共有フェーズ

33 使用フェーズ

300 第1の暗号デバイス

30

301 暗号システム

305 通信インターフェース

315 公開／秘密鍵生成器

325 生の鍵生成器

335 調整ユニット

340 カプセル化解除ユニット

350 第2の暗号デバイス

355 通信インターフェース

360 公開鍵取得器

365 公開／秘密鍵生成器

40

375 生の鍵生成器

385 調整データ生成器

390 カプセル化器

1000 コンピュータ可読媒体

1010 書き込み可能部分

1020 コンピュータプログラム

1110 集積回路

1120 処理ユニット

1122 メモリ

1124 専用集積回路

50

1 1 2 6 通信要素

1 1 3 0 相互接続部

1 1 4 0 プロセッサシステム

【発明を実施するための形態】

【0027】

本開示の主題は、様々な形態の実施形態を取り得るが、図面および以下の記載では、1つまたは複数の具体的実施形態が詳細に示される。本開示は、本開示の主題の原理の例示として考えられるべきであり、本開示の主題を図示および記載される具体的実施形態に限定するものではないことを理解されたい。

【0028】

以下、理解を目的として、動作中の実施形態の要素が説明される。しかしながら、各要素が、各要素によって実行されるものとして記載されている機能を実行するように構成されていることは明らかであろう。

【0029】

また、本開示の主題は実施形態に限定されず、本明細書に記載されている、または互いに異なる従属請求項に記載されている特徴は組み合わせることができる。

【0030】

以下に様々な実施形態を開示する。図1～図3には様々なプロトコルが示されており、第1の暗号デバイス10および第2の暗号デバイス20はこれらのプロトコルのために構成され得る。特に、第1の暗号デバイス10および第2の暗号デバイス20は鍵共有プロトコル、特に認証付き鍵共有プロトコル、より具体的にはパスワード認証付き鍵共有プロトコル(PAKE)のために構成され得る。

【0031】

実施形態は、いわゆるノイジーな乗算に基づき得る。例えば、秘密鍵から公開鍵を導出するために、第1の暗号デバイス10および第2の暗号デバイス20は、共通のオブジェクト(a)を秘密鍵と乗算し、ノイジーな乗算を使用して公開鍵を取得するように構成され得る。ノイジーな乗算の多くの例が知られており、例えばRound5、Saber、Kyber、NewHope、およびFrodoなどの暗号鍵共有プロトコルで使用されている。これらは、2つのパーティが共通の鍵について合意することを可能にする格子ベース鍵交換および／または鍵カプセル化メカニズムの例である。これらのスキームでは、従来は公開されるパラメータaを使用して、デバイス10およびデバイス20のそれぞれの秘密rおよびsから公開鍵コンポーネントbおよびuが取得され得る。少なくとも2つの鍵共有バリエーション、すなわちKEKプロトコルおよびKEMプロトコルが使用され得る。図1および図2では以下の表記法が使用されている。

— aは、第1の暗号デバイス10および第2の暗号デバイス20に知られている共通のオブジェクトを表す。例えば、共通のオブジェクトは、所与の環(例えば、RLWEまたはRLWRで使用される環)内の多項式、または整数エントリを有する行列(例えば、LWEまたはLWRで使用される行列)、または多項式エントリを有する行列であり得る。

— rおよびsはパーティ、例えばデバイス10およびデバイス20の秘密を表す。

— bおよびuはパーティの公開鍵または公開鍵シェアを表す。例えば、 $a * r$ または $a * s$ のノイジーな積として得られる。 $*$ 演算はノイジーな乗算を表し得る。ノイジーな乗算は、基礎となる暗号問題(例えば、(R)LWEまたは(R)LWR、あるいはそれらのモジュールバージョンなど)の一方向性関数であり得る。

【0032】

例えば、LWR $a * r$ の場合、 $\text{Round}((Ar \pmod q), p, q)$ として実装されてもよい。これは、qを法とする、長さnのベクトルであるrと $n \times n$ 正方行列Aとの積であり得る。そして、 $p/q(Ar) \pmod q$ を実行することによって結果が整数pおよびqを用いて丸められる($p < q$)。

— cはカプセル化されたデータを表す。

— hは調整ビット(reconciliation bits)を含むヘルパーデータを

10

20

30

40

50

表す。

- － 関数 s_d は、入力から最終シード τ を生成し、最終シード τ から共通のオブジェクト a を生成するように構成される。
- － 関数 g_{rc} は入力のための調整データを生成する関数である。例えば、関数 $g_{rc}(k)$ は、生の鍵 k から調整ビットを返すように構成され得る。関数 g_{rc} は $get_reconciliation(k)$ と称され得る。
- － 関数 rc は調整データを使用して生の鍵を調整するための関数である。例えば、 $rc(k', h)$ は、与えられた調整ビット h を用いて生の鍵 k を調整する関数であり得る。関数 rc は $reconciliate(k', h)$ と称され得る。
- － 関数 ecp は生の鍵を使用してデータをカプセル化するための関数である。例えば、 $ecp(k, K)$ は送信鍵 K が生の鍵 k でカプセル化されていることを意味し得る。例えば、カプセル化関数は、 k におけるエラーが m に対して及ぼす効果を限定するように（例えば、線形効果）、鍵 k を使用して鍵 K をマスクし得る。例えば、 k が Z_q 上の要素（例えば、ベクトル、行列、または多項式）である場合、鍵 K も Z_q において表され得る。カプセル化は成分ごとに実行され、例えば $c = k + K \cdot (q/2) \pmod{q}$ 。モジュラス q は偶数であってもよく、ある具体例では 2 の累乗であってもよい。送信鍵 K は、鍵共有プロトコルの結果として使用され得る鍵であってもよい。例えば、送信鍵は、その後のパーティ間のメッセージを保護するために使用されてもよい。さらなる保護は認証、および／または後のメッセージの暗号化であり得る。
- － 関数 dcp はカプセル化されたデータをカプセル化解除するための関数である。例えば、 $dcp(c, k')$ は生の鍵 k' を使用してデータ c のカプセル化を解除し、例えばビット列 K を返し得る。関数 dcp は $decapsulate(c, k')$ と称され得る。例えば、カプセル化を解除するために、 k' を差し引いて、 $q/2$ の最も近い倍数に丸めてもよい。

【0033】

一実施形態では、最終シード τ を導出するためにパーティに固有の追加情報が使用され、この情報はコンテキスト情報と呼ばれる。コンテキスト情報は秘密情報 p を含み得る。この秘密情報は、例えば、信頼できないパーティ、例えば第 1 の暗号デバイス 10 および第 2 の暗号デバイス 20 以外のパーティ、または第 1 の暗号デバイス 10 および第 2 の暗号パーティ 20 並びに製造業者等の様々な信頼できるパーティ以外のパーティに対して秘密であり得る。コンテキスト情報は秘密でなくてもよく、例えば公開であってもよい。コンテキスト情報を使用して最終シードを導出すると、通信チャネルの信頼性および鮮度の保証が向上する。鍵共有プロトコルにおいて共有される非秘密情報、例えばプレシード σ と、秘密ではないものの鍵共有プロトコル自体では共有されないさらなる情報 z は区別される。さらなる情報 z の例はネットワーク構成データやパブリック・ランダム・ビーコンなどを含む。

【0034】

例えば、パスワード認証付き鍵交換 (PAKE) を得るために、各パーティは秘密 p 、例えばパスワード、例えば PIN を共有してもよい。各パーティは通信リンクのために使用されるパラメータ a を $a = drbg(hash(\sigma, p))$ として計算し、ここで、 σ は公開シードとも呼ばれるプレシードであり、例えば、プロトコル中に平文でも交換され得るシードである。プレシードは最終シード τ を導出するための入力として使用され得る。共通オブジェクトは最終シード τ から導出され得る。

【0035】

この構造によれば、各パーティが同じ p を共有する場合、少なくとも暗黙認証付きの安全な接続が確立される。 s および r も秘密であるため、交換される公開コンポーネント b および u は p に関する情報を与えない。特に、 s および r がランダムに選択される場合、 b および u が p に関する情報を確実に与えない可能性がある。なお、鍵 K における機密性の保証は共通オブジェクトの秘密性に依存しないので、リンクのセキュリティは p 内の、例えば PIN 内のエントロピーの量に依存しない。このアプローチの利点は、(i) PA

K Eを得るために計算する必要のあるハッシュが少なくなること、および (i i) シードから共通オブジェクト a を例えば $a = \text{d r b g}(\text{s e e d})$ として計算する既存の N I S T P Q C の提案への組み込みが容易になることを含む。組み込みはシードが取得される方法を変更することを含み得る。

【 0 0 3 6 】

例えば、一実施形態では各パーティは次のように構成される。

ー 公開プレシード σ を取得する。例えば、第 1 のパーティはプレシードを、例えば乱数生成器を使用して生成し得る。第 1 のパーティはプレシードを第 2 のパーティに送り得る。この点において、プレシードは、暗号保護なしで共有され得るという意味で公開と見なされ得る。

ー 最終シード τ を算出する (例えば $\tau = \text{H a s h}(\sigma \mid p)$ 、例えば $\tau = \text{H a s h}(\sigma \mid p \mid z)$ など) ためのコンテキスト依存入力、例えば秘密 p (例えばパスワードおよび／またはさらなる情報 z) を取得する。ここで、 $\text{h a s h}()$ はハッシュ関数を示し、 $\mid$ は連結を示す。最終シード τ は、2つのパーティのそれぞれにおいて共通オブジェクト a を取得するために使用される。例えば、コンテキスト依存の共通オブジェクトは d r b g 、例えば決定論的乱数生成器、または P R N Gなどを適用することによって適用され得る (例えば、 $a = \text{d r b g}(\tau)$)。

【 0 0 3 7 】

興味深いことに、コンテキストに依存する (例えば、現在の鍵共有に固有の) 共通オブジェクトが使用され得る。これは、例えば事前共有秘密が使用される場合、暗黙認証を提供するために使用されてもよい。これは、中間者攻撃に対する耐性を高めるために使用され得る。これは、共通オブジェクトへの事前計算攻撃に対する耐性を高めるために使用され得る。

【 0 0 3 8 】

多くの異なる体または環における多くのノイジーな乗算が知られている。例えば、ノイジーな乗算は、真の乗算を実行し、ノイズ源を加えることによって実施されてもよい。例えば、ノイズは、例えばノイズ源から生成された明示的なノイズ、または例えば結果を丸めることによって生成された暗黙的なノイズであってもよい。例えば、 q を法とする乗算を丸めるために、 $\text{m o d } q$ 要素から $\text{m o d } p$ 要素にスケーリングされてもよく、例えば p / q を掛け、結果を丸めてもよい ($p < q$)。結果の丸めには様々な選択肢が存在し、例えば、最も近い整数への切り捨て、切り上げが挙げられ、丸めは、バイアスを減らす等のために値を加えることを含んでもよい。さらに、または代わりに、ノイズ源は乗算結果への明示的なノイズの追加、例えばガウスノイズの追加を含んでもよい。

【 0 0 3 9 】

例えば、第 1 および第 2 の公開鍵、第 1 および第 2 の秘密鍵、第 1 および第 2 の生の鍵、および共通オブジェクトは有限体または環上の行列またはベクトルであってもよい。例えば、第 1 および第 2 の公開鍵、第 1 および第 2 の秘密鍵、第 1 および第 2 の生の鍵、および共通オブジェクトは有限体または環上の多項式であってもよい。

【 0 0 4 0 】

鍵共有プロトコルにおける共通オブジェクト a を、各パーティによって共有されるコンテキスト情報にバインドすることにより、R o u n d 5、K y b e r、F r o d oなどの標準的な K E X プロトコルおよび K E M プロトコルにチャネル認証が導入され得る。チャネル認証は、M i t M またはなりすまし攻撃を回避するのに役立つ。

【 0 0 4 1 】

なお、一実施形態では、パーティによって使用されるパラメータ a は秘密であり得る。これは、共通オブジェクトが通常は公開パラメータである従来の鍵共有プロトコルとは異なる。有利なことに、これにより、暗号システムを破りたい攻撃者が利用できる情報がさらに減る。

【 0 0 4 2 】

図 1 は K E X プロトコルの実施形態の例を概略的に示す。図 1 ～図 3 では、図面の上か

10

20

30

40

50

ら下に向かって時間が増加する。

【0043】

図1のKEXにより、第1のパーティと第2のパーティが鍵を共有することができる。第1のパーティと第2のパーティは多数の調整ビットを交換するため、この鍵が失敗する可能性は低い。

【0044】

図1は、第2の暗号デバイス20と通信するように構成された第1の暗号デバイス10を示す。第1の暗号デバイス10は、暗号動作10.1を実行するように構成される。

【0045】

デバイス10の暗号動作10.1はプレシード σ を選択することを含み得る。例えば、プレシードはランダムに生成されてもよい。共通オブジェクトaは部分的にプレシードから導出されてもよい。したがって、共通オブジェクト内のエントロピーはプレシードを介して制御でき、例えば、共通のオブジェクトは少なくともプレシードと同程度にランダムであり得る。図1および図2ではランダム源は\$で表される。

【0046】

デバイス10の暗号動作10.1はプレシード σ および事前共有秘密pから最終シード τ を計算することを含み得る。2つ以上の入力、例えばこの場合はプレシード σ および事前共有秘密pからの最終シード τ の計算は、関数、例えばシード関数、例えばハッシュまたは鍵導出関数などを適用することによって実行されてもよい。

【0047】

例えば、事前共有秘密pは第2の暗号デバイスと第1の暗号デバイスとの間で事前共有されていてもよい。例えば、事前共有秘密pはアウトオブバンド通信チャネルを介してデバイス間で共有されてもよい。例えば、プロトコルは、第1の通信チャネル、例えば電子通信チャネル、例えば有線または無線通信チャネルを介してデバイス間で通信するように構成され得る。事前共有秘密pは異なる方法で、例えば第1の通信チャネルとは異なる第2の通信チャネルを介して伝送されてもよい。

【0048】

第2の通信チャネルの一実施形態では、秘密pは、例えば製造業者によって、第1のデバイス10および第2のデバイス20の一方または両方に事前にプログラムされている。例えば、秘密pは第1のデバイス10および第2のデバイス20の一方に事前にプログラムされていてもよく、または当該デバイスで生成されてもよい。例えば、ユーザがユーザインターフェースを介して第1および第2のデバイスのうちの他方のデバイスに秘密を入力してもよい。例えば、秘密は、ディスプレイに表示されてもよく、インターネットブラウザに表示されてもよく、またはデバイス上のステッカーに表示されていてもよく、その後、例えばユーザが、例えばユーザインターフェースを介して他方のデバイスに秘密pを入力する（例えばタイピングする）。複数の入力から、例えば、プレシード σ および共有秘密pから最終シード τ を計算するために、ハッシュ関数または鍵導出関数などが適用され得る。ハッシュ関数はさらなる情報zにも適用され得る。

【0049】

最終シードが計算されると、動作10.1は最終シード τ から共通オブジェクトaを計算することを含み得る。例えば、このために決定論的乱数生成器が使用されてもよい。例えば、共通オブジェクトaは、係数が最終シード τ からランダムに計算される行列または多項式であってもよい。図1では、最終シードの計算と共通オブジェクトの生成の組み合わせは関数sdにおいて組み合わせられている。例えば、関数sdは、例えば、プレシードと他の入力を組み合わせる最終シードにし、最終シードから共通オブジェクトを導出するために、ハッシュとdbrgの組み合わせとして実装されてもよい。

【0050】

一実施形態では、共通オブジェクト、公開鍵、および秘密鍵は1つまたは複数の多項式を含む。多項式の次数は、例えば少なくとも128、少なくとも256、少なくとも512などであり得る。共通オブジェクトおよび／または秘密鍵の係数は、少なくとも 2^8

10

20

30

40

50

、少なくとも 2^9 、少なくとも 2^{10} などである整数を法とし得る。秘密鍵の係数は、少なくとも 2^5 、少なくとも 2^6 、少なくとも 2^7 などの整数を法とし得る。

【0051】

動作10.1はさらに、第1の暗号デバイスに関連付けられた第1の秘密鍵 r を取得することを含み得る。

【0052】

秘密鍵はランダムに選択されてもよく、例えば、ランダムなソースから引き出されてもよい。これは、例えば、対応する公開鍵と共通オブジェクトとの間の接続をさらに不明瞭にすることによって、プロトコルのセキュリティを高める可能性がある。しかし、プロトコルごとに第1の秘密鍵 r などの秘密鍵をランダムに生成することは必須ではない。一実施形態では、鍵を計算するために、新しい秘密鍵が一方または両方のパーティによって使用される。しかし、秘密鍵を再利用することが可能であり、例えば、この特定の通信パーティ、またはプロトコルを実行し得る複数の暗号デバイスに関して再利用され得る。秘密鍵のサイズを小さくするために秘密鍵に様々な制限を課すことができ、例えば、秘密鍵内の非ゼロ係数の数が制限されてもよい。後者はオプションであり、必須ではない。

【0053】

第1の秘密鍵 r が取得された後、例えば生成された、または例えばデバイスのローカルストレージ等のストレージから取り出された後、動作10.1は、第1の秘密鍵 r から第1の公開鍵 b を計算することを含む。公開鍵を計算することは、秘密鍵に共通オブジェクト a を乗算すること、例えばノイジーな乗算を使用して乗算することを含む。

【0054】

第1の公開鍵 b およびプレシード σ は第2の暗号デバイス20に送られる。図1は、第1のデバイス10から第2のデバイス20への通信11.1を示す。通信11.1は第1の公開鍵 b およびプレシード σ を含み得る。

【0055】

図1は、第1の暗号デバイス10と通信するように構成された第2の暗号デバイス20を示す。第2の暗号デバイス20は暗号動作20.1を実行するように構成され得る。

【0056】

第2の暗号デバイス20は、暗号デバイス10から送られた第1の公開鍵 b およびプレシード σ を受け取るように構成される。第2の暗号デバイス20は、例えばデバイス20のストレージから、例えば上記と同じ方法で共有秘密 p を取り出すように構成される。

【0057】

動作20.1は、例えば、デバイス10と同じシード関数を適用することによって、第1の暗号デバイス10から受け取られたプレシード σ 、および事前共有秘密 p から最終シードを計算することを含み得る。動作20.1は、デバイス10およびデバイス20が同じ最終シード τ に到達するように構成され得る。動作20.1は、最終シードから同じ共通オブジェクト a を計算することを含み得る。動作20.1は、デバイス10およびデバイス20が同じ共通オブジェクト a に到達するように構成され得る。

【0058】

なお、共通オブジェクトは、第1のデバイス10と第2のデバイス20との間で通信チャネルを介して、例えば第1の通信チャネルを介して共有される必要はない。また、共有される情報、例えば σ 、 b は共通オブジェクトを計算するには十分ではない。これは、盗聴者が使用された共通オブジェクトを知らないことを意味する。

【0059】

動作20.1は、第2の暗号デバイスに関連付けられた第2の秘密鍵 (s) を取得することを含み得る。これを取得するための選択肢はデバイス10と同じであり、例えば生成されたり (例えばランダムに生成される)、例えば取り出される等である。第1の暗号デバイスに関連付けられた第1の公開鍵 b は第1のデバイスから取得され、例えば受け取られる。

【0060】

10

20

30

40

50

動作 20.1 は第 2 の秘密鍵 s から第 2 の公開鍵 u を計算することを含み得る。第 2 の秘密鍵 s は第 1 の公開鍵 b と同じ方法で計算され、例えば、第 2 の公開鍵 u を計算することは、第 2 の秘密鍵 s を共通オブジェクト a と乗算することを含み得る。

【0061】

動作 20.1 は、第 1 の公開鍵 b および第 2 の秘密鍵 s から、 k^* とも表記される第 2 の生の共有鍵 k を計算することを含み得る。第 2 の生の共有鍵の計算は、第 2 の秘密鍵 s と第 1 の公開鍵 b の乗算を含み得る。これは同じタイプの乗算であってもよいが、ノイズな乗算である必要はない。

【0062】

第 2 の公開鍵 u は第 1 のデバイス 10 に送信される。第 2 の公開鍵 u を使用して、第 1 のデバイス 10 は、第 2 の生の鍵とほぼ同じである第 1 の生の鍵を計算することができる。第 1 の生の鍵と第 2 の生の鍵が等しい場合もあるが、等しくない場合もある。第 1 の生の鍵と第 2 の生の鍵との間の潜在的な差異を減らすために様々な要素がプロトコルに追加され得る。例えば、図 1 は調整が使用される例を示す。

10

【0063】

例えば、一実施形態では、動作 20.1 は、生の共有鍵の係数の少なくとも一部のための調整データ h を生成することを含み得る。調整データ h は、第 1 および第 2 のデバイスにおいてそれぞれ導出された第 1 および第 2 の生の鍵の間の差異を低減することを可能にする情報を含み得る。

【0064】

最後に、動作 20.1 は、第 2 の生の共有鍵 k^* から送信鍵 K を生成することを含み得る。これにはハッシュ関数を適用することを含み得る。

20

【0065】

第 2 の公開鍵 u および調整データ h は第 1 の暗号デバイス 10 に送られる。図 1 は通信 21.1 を示す。例えば、通信 21.1 は第 2 の公開鍵 u および調整データ h を含み得る。

【0066】

第 1 の暗号デバイス 10 は、第 2 の暗号デバイス 20 から送られた第 2 の公開鍵 u および調整データ h を受け取るように構成される。第 1 の暗号デバイス 10 は暗号動作 10.2 を実行するように構成され得る。

【0067】

デバイス 10 の暗号動作 10.2 は、第 2 の公開鍵 u および第 1 の秘密鍵 r から第 1 の生の共有鍵 k' を計算することを含み得る。これは第 2 のデバイス 20 と同じ方法で行われてもよいが、使用される鍵は異なる。例えば、第 1 の生の共有鍵を計算することは、第 2 の公開鍵 (u) と第 1 の秘密鍵 (r) との間の乗算を含み得る。結果として、第 1 の生の鍵と第 2 の生の鍵は互いに近くなる可能性が高い。

30

【0068】

デバイス 10 の暗号動作 10.2 は、第 1 の生の共有鍵 k' 内の係数の少なくとも一部に対して、調整関数において調整データ h を適用することを含み得る。その結果、調整された第 1 の生の鍵は第 2 の生の鍵と等しくなる可能性が高い。そして、調整された第 1 の生の共有鍵 (k') から同じ送信鍵 K が得られ得る。生の鍵の係数の一部のみが調整される場合、第 1 および第 2 のデバイスは調整された係数を使用して送信鍵を計算し、調整されていない係数は無視してもよい。2 つのデバイスが同じ送信鍵に到達しない可能性がわずかに残っているが、この確率は管理可能であり、必要に応じて減らすことができる。

40

【0069】

暗号動作 10.1、20.1、および 10.2 はこの順番で実行されてもよい。

【0070】

調整データを計算する方法の例は、2016 年 11 月 4 日に EPO に出願された同出願人による出願番号 16197277.3 の特許出願、“REACHING AGREEMENT ON A SECRET VALUE” で説明されている。例えば、7~10 頁の方法が実施形態において調整のために使用され得る。この引用特許出願の他の箇所に開示され

50

ているバリエーションも採用可能である。本願では以下の3つの関数に次の表記法が採用される。

【0071】

1. 丸め関数

【数1】

$$\lfloor \cdot \rfloor_{B,b_h} : q, b_h, B \in \mathbb{Z}, b_h \geq 1, B < \log_2 q - b_h$$

に対して、

【数2】

$$\bar{B} = \log_2 q - B$$

10

とする。すると、

【数3】

$$\lfloor \cdot \rfloor_{B,b_h} : v \rightarrow \lfloor 2^{-\bar{B}} \cdot v \rfloor \pmod{2^B}$$

直感的には、

【数4】

$$\lfloor v \rfloor_{B,b_h}$$

20

は、

【数5】

$$\{v + 2^{\log q - (B+b_h)}\}$$

のB個の最上位ビットを抽出する。ここで、2番目のコンポーネントは、バイアスのない丸め誤差を保証するための丸め係数である。Bは記号vから抽出されるビットの数を示し、b_hはヘルパーデータビットの数を示す。一実施形態ではqは2の累乗であり得る。

30

【0072】

2. クロス丸め (cross-rounding) 関数

【数6】

$$\langle \cdot \rangle_{B,b_h} : q, b_h, B \in \mathbb{Z}, b_h > 1, B < \log_2 q - b_h$$

に対して、

【数7】

$$\bar{B} = \log_2 q - B$$

40

とする。すると、

【数8】

$$\langle \cdot \rangle_{B,b_h} : v \rightarrow \lfloor 2^{-\bar{B}+b_h} \cdot v \rfloor \pmod{2^{b_h}}$$

直感的には、

【数9】

50

$\langle v \rangle_{B, b_h}$

は、 v の $(B + b_h)$ 個の最上位ビットの b_h 個の最下位ビットを抽出する。

【0073】

3. 調整関数 $rec(w, b)$:

【数10】

$q, b_h, B \in \mathbb{Z}, b_h \geq 1, B < \log_2 q - b_h, w \in \mathbb{Z}_q, b \in [0, 2^{b_h}),$

10

に対して、

【数11】

$rec(w, b) = \lfloor v \rfloor_{B, b_h}$

ここで、 v は

【数12】

$\langle v \rangle_{B, b_h} = b$

20

であるような w に最も近い要素である。最も近い要素 w はリー距離に応じて取得され、例えば $\min(|v - w|, q - |v - w|)$ である。

【0074】

これらの3つの関数は多項式や行列などに対して係数ごとに適用されてもよい。本明細書では上記調整関数を例として使用する。前述したように、上記の引用出願内の調整方法を使用することも可能である。クロス丸め関数は調整データを取得するために適用され、丸め関数は、調整されるデータを取得するために適用され得る。調整データを後に調整関数において使用することで調整されたデータを復元することができる。言い換えれば

【数13】

$rec(w, \langle v \rangle) = \lfloor v \rfloor$

30

であり、ここで、 v および w は互いの閾値距離内にあるものと仮定される。

【0075】

一実施形態では、第1および第2の公開鍵、第1および第2の秘密鍵、および生の鍵は、有限体または環上の複数の多項式であり、ここで公開鍵は、複数の共有される多項式 (a) とのノイズな乗算によって秘密鍵から取得される。例えば、複数の多項式は、格子の要素が多項式である複数のモジュラー格子において使用され得る。

【0076】

図2はKEMプロトコルの実施形態の例を概略的に示す。図2に示されるプロトコルは図1のプロトコルと概して同じであるが、生の鍵がカプセル化に使用される点、および生の鍵間で生じ得る差異の確率が異なる方法で低減される点で異なる。図2のプロトコルの利点は、第2のパーティは、第1のパーティの公開鍵 b に基づき、自身が選択したランダムキーをカプセル化できることである。これによりアクティブなセキュリティを実現することができる。

40

【0077】

図1に係る実施形態では、第1のデバイス10が最初に暗号動作を実行し、これらは図1に示される動作10.1の実施形態であり得る。デバイス10からデバイス20への通信11.1は図1と同じであってもよい。

【0078】

50

図2はその後、プレシード σ および公開鍵 b を受け取った後に暗号動作20.2に進み得る。暗号動作20.2は、調整データが計算されないことを除いて、動作20.1と同じであってもよい。代わりに、動作20.2は送信鍵 K の生成を含み得る。図1と図2の違いは、図2では最終的な送信鍵がどうなるかを制御でき、例えば、第2のデバイス20がそれを選択し得る。選択はランダム生成であってもよいが、別のアルゴリズムの結果であってもよい。

【0079】

調整データの代わりに、カプセル化関数（例えば、関数 ecp ）を適用することによって、第2の生の共有鍵の少なくとも一部を用いて送信をカプセル化して、カプセル化されたデータ c が取得される。

【0080】

カプセル化されたデータ c は、例えば第2の公開鍵 u とともに第1の暗号デバイスに送られる。例えば、第2のデバイス20は第1のデバイス10に通信21.2を送ってもよい。

【0081】

第1のデバイス10は、例えば第2の公開鍵 u とともに第2の暗号デバイス20から送られたカプセル化されたデータ c を受け取るように構成され得る。第1のデバイス10は暗号動作10.3を実行するように構成され得る。動作10.3は動作10.2と類似しているが、第1の生の鍵と第2の生の鍵との間の差異を減らすために調整の代わりにカプセル化が使用される点で異なる。

【0082】

例えば、動作10.3は、第1の生の共有鍵 k' の少なくとも一部を使用してカプセル化されたデータ c のカプセル化を解除することで送信鍵 K を取得することを含み得る。

【0083】

一実施形態では、送信鍵 K はランダムであり、かつ／または一時的であり、かつ／または対称であり、かつ／または第1の公開鍵（ b ）から独立している。

【0084】

例えば、送信鍵 K は、第1の秘密鍵、第1の公開鍵、第2の秘密鍵、および第2の公開鍵から独立して生成され、すなわち、送信鍵 K は第1および第2の生の鍵から独立していてもよい。なお、生の鍵は公開鍵および秘密鍵に依存する。例えば、送信鍵は少なくとも部分的にランダムに生成されてもよい。独立した鍵はアクティブなセキュリティの実現をより容易にする。鍵、例えば秘密鍵または送信鍵を計算するためのランダム性は、従来の乱数生成器、好ましくは真の乱数生成から取得され得る。

【0085】

一実施形態では、共通オブジェクトおよび秘密鍵は q を法とする係数を有する次数 n の多項式である。公開鍵は p を法とする次数 n の多項式であってもよい。

【0086】

広範囲のパラメータが可能であるが、一般に、得られる鍵共有プロトコルが、十分に低い失敗の確率で十分に大きな鍵を生成するという点で制限される。後者の2つのパラメータは経験的に観測されてもよいし、あるいは理論的に計算されてもよい。許容値は用途に依存する。

【0087】

図2に関連して使用され得る特定の有利なKEMプロトコルは、参照によって本明細書に援用されるNISTのプロポーザル“Round 5: KEM and PKE based on (Ring) Learning with Rounding Thursday 28th March, 2019”に記載されている。例えば、セクション2.9.5および2.9.7には幅広い状況に適したパラメータの例が示されている。例えば、上記文献はカプセル化、エラー訂正、およびその他のKEMプロトコルの詳細の例を提供する。

【0088】

10

20

30

40

50

例えば、例えば図2のようなKEMプロトコルは、好ましくは $n+1$ の素数（例えば、 q を法とする係数を備えた次数 n の多項式）を有する円分環 $Z_q[x]/\Phi_{n+1}(x)$ 上で定義され得る。還元多項式(reduction polynomial)は任意のものであり得るが、 $(n+1)$ 番目の円分多項式 $\Phi_{n+1}(x) = x^n + \dots + x + 1$ が効率的な選択であり得る。共通オブジェクト A は $Z_q[x]/\Phi_{n+1}(x)$ 内のエントリを有する $d/n \times d/n$ 行列であり得る。秘密鍵は

【数14】

$$d/n \times \bar{n}$$

10

または

【数15】

$$d/n \times \bar{m}$$

のサイズを有する行列であってもよい。

【0089】

例えば、以下のパラメータが使用されてもよい。

d ：公開行列の行ごとの多項式係数の数

n ：還元多項式の次数。また、環($n > 1$)のインスタンス化または環ではない($n = 1$)インスタンス化のどちらが使用されているかを示す。

20

h ：秘密行列の列ごとの非ゼロ多項式係数の数

q 、 p 、 t 、 b ：丸めモジュラス、例えば $b \mid t \mid p \mid q$ 、例えば $b < t < p < q$ を満たす全ての2の累乗

b_bits ：暗号文コンポーネント内のシンボルごとに抽出されたビットの数、 $= 2^{b_bits}$

κ ：セキュリティパラメータ、送信鍵の情報ビットの数

【数16】

$$\bar{n}$$

30

：イニシエータの秘密行列の列数

【数17】

$$\bar{m}$$

：レスポндаの秘密行列の列数

【0090】

例えば、一実施形態では $d = 618$ 、 $n = 618$ 、 $h = 104$ 、 $q = 2^{11}$ 、 $p = 2^8$ 、 $t = 2^4$ 、 $b = 2^1$ 、 $\kappa = 128$ 、

40

【数18】

$$\bar{n} = 1$$

、

【数19】

$$\bar{m} = 1$$

が選択され得る。

50

【0091】

エラーの確率をさらに下げるために、カプセル化の前に送信鍵を符号語に符号化してもよい。これは以下の追加パラメータを使用し得る。

f：エラー訂正コードによって訂正可能なビットエラーの数

xe：エラー訂正コードのパリティビット数

μ：暗号文コンポーネント

【数20】

$$\mu = \left\lceil \frac{\kappa + xe}{b_bits} \right\rceil$$

10

内のシンボルの数

【0092】

例えば、上記の場合、上記パラメータは $f = 0$ 、 $xe = 0$ 、 $\mu = 128$ であってもよい。一実施形態では

【数21】

$$\mu \leq \bar{n} \cdot \bar{m} \cdot n$$

かつ $\mu \cdot b_bits \geq \kappa$ であり得る。

20

【0093】

別のインスタンス化は

$d = 1186$ 、 $n = 1$ 、 $h = 712$ 、 $q = 2^{15}$ 、 $p = 2^{12}$ 、 $t = 2^7$ 、 $b = 2^4$ 、 $\kappa = 256$ 、

【数22】

$$\bar{n} = 8$$

、

【数23】

$$\bar{m} = 8$$

30

、 $f = 0$ 、 $xe = 0$ 、 $\mu = 64$ であり得る。

【0094】

別のインスタンス化は

$d = 490$ 、 $n = 490$ 、 $h = 242$ 、 $q = 2^{10}$ 、 $p = 2^7$ 、 $t = 2^3$ 、 $b = 2^1$ 、 $\kappa = 128$ 、

【数24】

$$\bar{n} = 1$$

40

、

【数25】

$$\bar{m} = 1$$

、 $f = 5$ 、 $xe = 190$ 、 $\mu = 318$ であり得る。

【0095】

同様のパラメータが図1に係るプロトコルに使用されてもよい。

50

【0096】

第1の生の鍵と第2の生の鍵との間の距離を短くするための他の様々な方法が存在し、図1および図2に関連して示されたものに加えてまたは代わって使用され得る。例えば、一実施形態ではカプセル化と調整の両方が使用される。その場合、カプセル化に調整された生の鍵が使用され得る。例えば、調整および／またはカプセル化を計算する前に、第2の生の鍵のいわゆる高信頼性ビット、例えば第1のデバイス10で同じビットを生成する可能性が最も高い第2の生の鍵の係数が選択され得る。その後、調整および／またはカプセル化が選択された高信頼性係数に適用され得る。高信頼係数のインデックスは、例えば通信21.2においてデバイス20からデバイス10に伝えられ得る。

【0097】

最終シード τ 、プレシード σ 、および／または事前共有秘密 p の様々な計算方法の例を以下に示す。これらの例はそれぞれ、本明細書に記載されている様々な鍵共有プロトコル、特に図1および図2のプロトコルに組み込まれ得る。

【0098】

例えば、一実施形態では、事前共有秘密は事前共有パスワードを含み得る。例えば、パスワードは英数字のシーケンスまたは数字のシーケンスであってもよい。例えば、事前共有秘密はアウトオブバンドチャネルを介して取得され得る。

【0099】

例えば、一実施形態では、プレシード σ は平文形式で送受信され得る。例えば、デバイス10は、プレシード σ をランダムに生成し、平文形式で、例えば暗号化せずに第2のデバイス20に送ってもよい。興味深いことに、システムのセキュリティはこれによる影響を受けない。攻撃者はプレシード σ のみから共通オブジェクト a を取得できず、さらに、仮にできたとしても、結果として得られる送信鍵 K システムのセキュリティは共通オブジェクト a の秘密性に依存しない。

【0100】

例えば、プレシードは、第1のデバイス10および第2のデバイス20に設けられ、鍵共有プロトコルの通信（例えば、メッセージ）、および／または両デバイス間で送信鍵で保護された状態で交換され得る後続の通信を交換するために使用される通信インターフェースを介して送受信され得る。一実施形態では、プレシードはこれらの通信インターフェースを介して送られない。

【0101】

例えば、最終シード τ は事前シードおよび事前共有秘密以外のさらなる情報に依存し得る。例えば、さらなる情報はハッシュまたは鍵導出関数に含まれてもよい。例えば、最終シードは、第2の暗号デバイスと第1の暗号デバイスとの間のネットワーク接続からのネットワーク構成データからさらに計算され得る。ネットワーク構成データが公開データであったとしても、攻撃者はさらなる情報を追跡する必要があるため、MitMまたはなりすまし攻撃を緩和するのに有用である。

【0102】

例えば、さらなる情報は第1および／または第2の暗号デバイスのネットワークアドレスを含み得る。ネットワークアドレスの例はIPアドレスまたはMACアドレスである。

【0103】

例えば、さらなる情報 z はネットワーク接続の目的を含んでもよい。例えば、ネットワーク接続の目的は、特定のアプリケーションのための通信を実行することであり得る。アプリケーションの名前がさらなる通信に含まれ得る。接続の目的は、以前に導出された情報から導き出されてもよい。例えば、ネットワークポートアドレスを使用して目的が導き出されてもよい。ネットワークポート自体もさらなる情報の一部であってもよい。例えば、鍵共有プロトコルは、その後に実行されるさらなるアルゴリズムを理由として実行されてもよい。さらなるアルゴリズムの指標は、最終シードを計算するためのさらなる情報として使用され、後に、さらなるアルゴリズムが実際に使用されるか否かを確認するため、またはさらなる情報と一致しないアルゴリズムを拒否するために検証され得る。

10

20

30

40

50

【0104】

例えば、さらなる情報 z は時刻を含み、場合によっては現在の日付を含み得る。例えば、第1および第2のデバイスはまず、各々の時間を同期させるために同期プロトコルを実行し、その後、同期された時間をさらなる情報として使用してもよい。例えば、現在の日付がさらなる情報に含まれてもよい。

【0105】

例えば、さらなる情報 z はインタラクションカウンタを含み得る。インタラクションカウンタは第2の暗号デバイス20および第1の暗号デバイス10によって維持され得る。例えば、インタラクションカウンタは、少なくとも共有鍵が導出されるときには常に増加してもよい。

10

【0106】

したがって、3つのソース、すなわち（例えば、通信11.1において）共有され得るプレシード σ 、事前共有秘密 p 、およびさらなる情報から最終シード τ を導出することによって、攻撃に対するさらなる耐性が得られる可能性がある。実際には、2つのソースの別の選択、例えばプレシード σ およびさらなる情報 z 、または例えば事前共有秘密 p およびさらなる情報から最終シード τ を導出することによっても、異なる性質を持つ有益なプロトコルが得られる。

【0107】

さらなる情報で使用されるネットワーク構成データは、第1の暗号デバイスと第2の暗号デバイスとの間の現在のネットワーク接続からのものであってもよく、かつ／または、さらなる情報で使用されるネットワーク構成データは、第1の暗号デバイスと第2の暗号デバイスとの間の以前のネットワーク接続からのものであってもよい。

20

【0108】

例えば、一実施形態では、事前共有秘密が合致する場合、鍵共有が成功する。例えば、共有秘密 p はPINまたは事前共有鍵（PSK）であり得る。各パーティが事前共有秘密（例えば、PINまたはPSK）を知っている場合、両者は最終シード τ を導出でき、そしてそこから共通オブジェクト a を導出できる。例えば、 $\tau = \text{Hash}(p \parallel \sigma)$ および $a = \text{drgb}(\tau)$ 。最終シード τ が取得された後は、シードを使用する既知のKEXまたはKEMプロトコルを通常どおり実行することができる。例えばMitMやなりすましなどの攻撃が行われている場合、KEXまたはKEMプロトコルは共有対称鍵に導いたり、適切なカプセル化解除を許容したりしない。確立された鍵の後の通常使用は、両パーティが同じ秘密 p を共有したか否かを暗黙的に決定する。暗黙認証は明示認証に変換され得る。例えば、通信11.1はチャレンジを含み、通信21.1または21.2は対応するレスポンスを含み得る。例えば、チャレンジはナンスであり、レスポンスは送信鍵を用いた暗号化であってもよい。送信鍵を導出した後、デバイス10はレスポンスを検証することで、デバイス20が自身のものと合致する秘密 p を有するか否かを知り得る。チャレンジは暗黙的であってもよく、例えば、チャレンジは所定のストリングであってもよい。

30

【0109】

図1および図2のメッセージまたはその後のメッセージは、メッセージ認証コードまたはメッセージ認証タグを含み得る。それらが正しく検証できれば、暗黙認証を明示認証に変換できる。それらが正しく検証できない場合、暗黙的な認証の欠如を明示的な認証の欠如に変換できる。

40

【0110】

なお、短い共有秘密が使用される場合、例えば、可能な値を10000しか有さない4つの数字からなるPINが使用される場合であっても、この構造はPINを簡単にブルートフォース攻撃することはできないことを保証する。その理由は、秘密の値 p の数が制限されている（10000）場合であっても、この値は接続を通過しないため、攻撃者は使用されるPINを導出できない。接続を介して交換される値、すなわち公開鍵コンポーネント b および u は、共通オブジェクト a を s および r と乗算することによって取得される。 s および r はともに秘密であることから、これらの公開鍵コンポーネントを共有秘密に

50

簡単に関連付けて復元することはできない。対照的に、従来のアルゴリズムではパスワードのハッシュが接続を通過し、パスワードのエントロピーが低い場合、パスワードのリークに直接つながるおそれがある。もう1つの利点は、どちらのパーティも相手の共有秘密を学ばないことである。なぜなら、公開鍵の知識から共通オブジェクトを導出することができないからである。

【0111】

一実施形態では、最終シード τ はさらなる情報 z から、場合によってはプレシード σ および事前共有秘密 p に加えてさらなる情報から導出される。この構造は、パーティ間の安全な接続のセットアップにおいて、例えば $IPSec$ などのネットワークプロトコルを用いて利用され得る。例えば、一実施形態では各パーティは定期的に接続を確立し、その場合、さらなる情報は例えば以下の情報を含み得る。

- (1) 第1のパーティの現在のIPアドレス、(2) IP接続の目的(例えば、後に使用される他のプロトコル)、(3) 時刻を含む現在の交換に関する情報。
- 以前の交換に関する情報、例えば、(1) 鍵交換が最後に行われたのはいつか、(2) 交換の目的。この情報は公開されている可能性があるが、攻撃者がなりすまし攻撃または $MitM$ 攻撃でなりすましに成功するには、両パーティ間の状態を追跡する必要があり得ることを意味する。
- 乱数ビーコンからの乱数(例えば、 $NIST$ ランダムビーコン)。最後に開示されたビーコンの乱数がさらなる情報において使用される場合、第2のパーティはこれが新しいハンドシェイクであって、リプレイではないことがわかる。
- 現在の接続に関する情報(例えば、IPアドレス)。その場合、対応するパーティが鍵交換をその特定のIPアドレスにバインドする。これにより、通信の一部が別のコンテキストで再利用されるミックスアンドマッチ攻撃を回避できる。
- 第1のパーティが後に使用する予定のプロトコルに関する情報が含まれる。その場合、第2のパーティは目的が適切か否かを早期に確認できる。早期確認ができるので、DOS攻撃がより困難になる。
- 以前の鍵交換に関する情報。これにより、攻撃者は、プロトコルを破るには過去の交換を追わなくてはならない。

【0112】

興味深いことに、事前共有秘密 p が使用される場合、攻撃者は公開コンポーネント b および u に基づき秘密コンテキスト p を復元することはできない。実際には、 p のスペース、例えばセキュリティパラメータのサイズが大きい場合、セキュリティは、ハッシュ関数の出力のプレイメージを見つけることと同等である。したがって、安全なハッシュ関数が使用される場合、システムは安全である。しかし、秘密 p が限られた数の状態しか有さない、極端な例では2つの可能な状態しか有さない場合であっても、セキュリティは次の問題と同等である。公開 $(a_1, a_2, b) = (\text{hash}(\sigma, m), \text{hash}(\sigma, 1-m), m \cdot A_1 * s + (1-m) \cdot A_2 * s)$ および公開シード、秘密 $m=0$ または 1 、および秘密 s を使用して、 m を求める、言い換えれば a_1 と a_2 のどちらが使用されたかを突き止める。この問題では、 A_1 および A_2 はシードから既知の方法で導出可能な共通オブジェクトであり、例えば $A_1 = \text{sd}(\sigma, m)$ 等が使用され得る。

【0113】

LWE/R 仮定は、一様ランダムから $(A, b = A * s)$ を区別することは困難であると述べている。上記の問題では、 $(A_1, b = A_1 * s)$ および $(A_2, b = A_2 * s)$ はともに一様から区別するのが困難な、よって、互いから区別するのが困難な LWE/R サンプルである。

【0114】

図3は、第1のデバイス10と第2のデバイス20との間のプロトコルの一実施形態の例を概略的に示す。

【0115】

図3には、事前共有秘密 p を共有するためのアウトオブバウンド通信13が示されてい

10

20

30

40

50

る。アウトオブバウンド通信 13 はデバイス 10 および 20 の登録フェーズ 31 中に行われ得る。例えば、アウトオブバウンド通信 13 は、例えば後の通信を可能にするために、デバイスの製造中に実行され得る。例えば、アウトオブバウンド通信 13 は、例えば後の通信を可能にするために、デバイスの製造中に部分的に実行されてもよい。他方のデバイスには後に事前共有秘密が提供されてもよい。例えば、アウトオブバウンド通信はユーザによって、例えば一方または両方のデバイス上のユーザインターフェースを介して手動で実行され得る。例えば、事前共有秘密は、デバイスのスイッチ（例えば、ディップスイッチ）を設定することによって、または別のインターフェースを介して秘密を表すことによって、一方のデバイス、例えばデバイス 20 に設定され得る。秘密は他方のデバイスで生成され、他方のデバイスで設定されてもよい。

10

【0116】

後の鍵共有フェーズ 32 において、第 1 のデバイス 10 および第 2 のデバイス 20 は一実施形態（例えば、図 1 または図 2 等に関して本明細書に記載される実施形態）に係る鍵共有プロトコルを実行する。鍵共有プロトコルでは 2 つのメッセージ、すなわち、（例えば本明細書に記載されるような）鍵共有プロトコルメッセージ 14 および鍵共有プロトコルメッセージ 15 が交換され得る。鍵共有プロトコルにはより多くのメッセージが含まれていてもよい。鍵共有プロトコルにおいて事前共有秘密 p が使用され得る。

【0117】

鍵共有フェーズ 32 の終了時、両デバイスは、秘密 p に関して少なくとも暗黙的に認証された送信鍵 K を有し得る。デバイス 10 とデバイス 20 が同じ共有を有さない場合、少なくともほとんどの場合、2 つのデバイスで導出される送信鍵 K は異なる。

20

【0118】

一実施形態では、登録フェーズ 31 および鍵共有フェーズ 32 は異なる通信チャネル、例えば異なる通信媒体を使用するが、異なる周波数または異なる周波数帯域を使用してもよい。

【0119】

後の使用フェーズ 33 において、鍵共有フェーズ中に合意された送信鍵を使用され得る。例えば、送信鍵 K を使用して暗号的に保護されたメッセージがデバイス 10 とデバイス 20 との間で交換され、例えば、メッセージは鍵 K で暗号化されてもよく、または、鍵 K を用いて計算された認証タグ（例えば MAC ）を含んでもよく、あるいは両方であってもよい。メッセージは鍵 K から導出された鍵を用いて保護されてもよい。図 3 は 2 つのメッセージ 16 および 17 を示しているが、2 つ以上のメッセージ、例えば 4 つ以上、または 1 つのメッセージが存在してもよい。メッセージの順序は異なる順序であり得る。

30

【0120】

例えば、デバイス 20 は、メッセージ m を送信鍵 K を用いて暗号化し、暗号化されたメッセージを第 1 の暗号デバイス 10 に送るように構成されてもよい（またはその逆）。デバイス 10 は送信鍵 K を使用してメッセージを復号するように構成され得る。

【0121】

送信鍵は生の共有鍵から導出された鍵である。例えば、送信鍵は、例えば KEM プロトコルにおいて、生の共有鍵から独立していてもよい。一実施形態では、暗号化に使用される鍵は送信鍵および公開鍵のハッシュであり得る。例えば、受信者の公開鍵、送信者の公開鍵、または両方を使用してハッシングしてもよい。例えば、送信鍵は生の共有鍵から、例えば生の鍵をハッシングすることによって、直接的に導出されてもよい。一実施形態では、暗号化に使用される鍵は生の鍵および公開鍵のハッシュであり得る。

40

【0122】

鍵共有フェーズ 32 または使用フェーズ 33 において、暗黙認証が明示認証に変換されてもよい。例えば、チャレンジ／レスポンスプロトコルは送信鍵を使用して、例えば送信鍵を直接使用して、または送信鍵から導出された鍵を使用して行われてもよい。

【0123】

興味深いことに、チャレンジ／レスポンスプロトコルは暗黙的であり得る。例えば、使

50

用フェーズ 3 3 中、保護の一部としてメッセージが認証され得る。認証が失敗した場合、これはメッセージが改ざんされているか、または事前共有秘密 p がデバイス間で合致しなかったことを意味する可能性がある。いずれにせよ、通信は信頼できず、適切な措置が取られ得る。問題を検出した際の適切な措置は、例えば、通信の中止、新しい鍵共有プロトコルの開始、警告メッセージ（例えば、電子メールやアラームなど）の送信を含み得る。

【0124】

例えば、使用フェーズ 3 3 中、保護の一部としてメッセージが暗号化され得る。メッセージが予測可能な部分、例えば、事前に決定されたストリングやヘッダー（例えば、カウンタ）などを含む場合、復号後に予測可能な部分が検証されてもよい。予測可能な部分が復号後の予測と合致しない場合、メッセージが改ざんされているか、または事前共有秘密 p がデバイス間で合致しなかったと結論付けられ得る。この場合、適切な措置が取られ得る。

【0125】

興味深い適用例は、デバイスに、例えばネットワーク構成データを提供することである。例えば、ネットワーク通信を実行するように構成されたデバイスを想定する。例えば、ネットワーク通信のためのデバイスを構成するように構成されたデバイスを想定する。例えば、一実施形態では、ネットワーク通信を実行するように構成されたデバイスはデバイス 20 であり、構成デバイスはデバイス 10 であり得る。ただし、この逆であってもよい。構成デバイスはスマートフォン、ラップトップ、コンピュータなどであってもよい。

【0126】

例えば、登録フェーズ 3 1 のために、デバイス 20 は事前共有秘密を受け取り得る。事前共有秘密はデバイス 10 にアウトオブバウンド伝送される。例えば、事前共有秘密は、デバイスの外部で提供され、ユーザによって入力されてもよく、またはスキャンされてもよい（例えば、QR コード（登録商標）やバーコード等をスキャンする形式で）。事前共有秘密がデバイス 10 において、例えば適切なインターフェース（例えば、ユーザインターフェースまたはスキャンインターフェース）を介して利用可能である場合。例えば、事前共有秘密は NFC または Bluetooth（登録商標）を使用して共有されてもよい。例えば、デバイス 10 および／またはデバイス 20 上のボタンを押すことによって交換が開始されてもよい。例えば、事前共有秘密はデバイス 10 および／またはデバイス 20 上でランダムに生成されてもよい。

【0127】

鍵共有フェーズ 3 2 において、デバイス 10 およびデバイス 20 は事前共有秘密を使用して送信鍵 K について合意する。例えば、鍵共有フェーズは Wi-Fi（登録商標）などの別の通信チャネルを使用してよい。

【0128】

使用フェーズ 3 3 中、デバイス 10 は送信鍵を使用して保護された（例えば、暗号化された）ネットワーク構成データをデバイス 20 に送信してもよい。例えば、送信は、鍵共有フェーズ 3 2 と同じ通信チャネル、例えば Wi-Fi（登録商標）などを使用し得る。ネットワーク構成データを受信した後、デバイス 20 はネットワーク構成データを使用して、コンピュータネットワークを介してデータを送受信し得る。ネットワーク構成データはネットワークキー、証明書、ネットワークアドレスなどを含み得る。送信鍵を用いたメッセージ（例えば、ネットワーク構成データ）の暗号化は、送信鍵を用いた直接的な暗号化、およびメッセージを直接的に暗号化するために使用される別の鍵を送信鍵から導出する間接的な暗号化を含む。例えば、デバイス 20 はルーターであってもよい。上記のようなプロトコルは照明要素、例えば照明器具を構成するために使用されてもよい。

【0129】

興味深いことに、実施形態は、例えば通信を保護するために事前共有秘密を直接使用するよりも安全な送信鍵を提供し得る。事前共有秘密のエントロピーが低い場合でも、実施形態によって提供される送信鍵は高いエントロピーを有することができる。

【0130】

例えば、安全で認証付きの通信リンクを確立するために Bluetooth（登録商標）デバイスのペアが結合され得る。リンクは、鍵共有プロトコルによって提供される機密性強度の点で安全であり、また、両者が同じ事前共有秘密を使用することから認証付きである。

【0131】

一実施形態はなりすまし攻撃に対処するために使用されてもよい。例えば、共通オブジェクトがプロトコルの一部として固定または共有されている従来のプロトコルでは、攻撃者は一方からのメッセージをリプレイできる可能性がある。第2のパーティが何かがおかしいことに気付くまでに長い時間がかかるおそれがある。しかし、第2のパーティがメッセージを受信した際、第2のパーティは受信されたメッセージが本当に特定のソースからのものであることを確認したい可能性がある。一実施形態では、メッセージに変化する要素を含めることによってこれが達成され得る。例えば、送信鍵を用いた暗号化または認証のために、第2のパーティはチャレンジ、例えばナンスをメッセージ内に、例えばメッセージ21.1または21.2内に含めて他方のパーティに送信し得る。

【0132】

図4は、一実施形態に係る暗号システム301の例を概略的に示す。システム301は第1の暗号デバイス300および第2の暗号デバイス350を含む。例えば、第1のデバイス300および／または第2のデバイス350などのデバイス上の第1のデバイス300および第2のデバイス350であってもよい。例えば、デバイス300はデバイス10として構成され、デバイス350はデバイス20として構成され得る。

【0133】

第1および第2のデバイス300および350は暗号プロトコルを実行するように構成される。両デバイスは一方のデバイスから他方のデバイスにデータを渡す機能を備える。

【0134】

例えば、第1および第2のデバイスは鍵共有プロトコルのために構成され、例えば、両デバイス間で共有される鍵、典型的には対称鍵を生成するように設計され得る。その後、共有鍵は、保護された通信、例えば暗号化かつ／または認証された通信のためにデバイスによって使用され、例えば、メッセージを暗号化するために、および／またはメッセージのための認証タグを計算するために使用され得る。

【0135】

一実施形態では、デバイス300およびデバイス350はそれぞれ事前共有秘密pを保存していてもよい。例えば、デバイス300およびデバイス350は、事前共有秘密を取得および／または保存するための事前共有秘密ユニット（別個に図示されていない）を含み得る。

【0136】

第1のデバイス300および第2のデバイス350はストレージインターフェース、プロセッサ、およびメモリのうちの1つまたは複数を含み得る。第1のデバイス300および第2のデバイス350（例えば、システム301の様々なデバイス）は、コンピュータネットワーク391を介して互いに通信することができる。コンピュータネットワークはインターネット、イントラネット、LAN、WLAN等であり得る。コンピュータネットワーク391はインターネットであり得る。コンピュータネットワークは全体的または部分的に有線であってもよく、かつ／または全体的または部分的に無線であってもよい。例えば、コンピュータネットワークはイーサネット接続を含み得る。例えば、コンピュータネットワークはWi-Fi（登録商標）やZigBee（登録商標）などの無線接続を含み得る。デバイスは、必要に応じてシステム301の他のデバイスと通信するように構成された接続インターフェースを備える。例えば、接続インターフェースはコネクタ、例えば、有線コネクタ（例えば、イーサネットコネクタ）または無線コネクタ（例えば、アンテナ（例えば、Wi-Fi（登録商標）、4Gまたは5Gアンテナ））を含み得る。例えば、第1のデバイス300および第2のデバイス350はそれぞれ通信インターフェース305、355を含み得る。コンピュータネットワーク391は追加の要素、例えばルー

ターやハブなどを含み得る。

【0137】

第1のデバイス300および第2のデバイス350の実行はプロセッサ、例えばプロセッサ回路において実装され、プロセッサの例は本明細書に示される。第1のデバイス300、特に第1のデバイス300のプロセッサは第1のパーティ10の機能を実装し得る。第2のデバイス350、特に第2のデバイス350のプロセッサは第2のパーティ20の機能を実装し得る。例えば、これらの機能は、デバイス300または350に保存された（例えば、デバイスの電子メモリ内に保存された）コンピュータ命令として完全にまたは部分的に実装され得る。コンピュータ命令はデバイスのマイクロプロセッサによって実行可能である。ハイブリッド実施形態では、機能ユニットは部分的にハードウェアで、例えばコプロセッサ、例えば暗号化コプロセッサとして実装され、部分的に、デバイス300または350上で格納および実行されるソフトウェアで実装される。

10

【0138】

デバイス300および350は、メッセージ（場合によっては暗号化されたメッセージ）を保存および／または取り出すためのストレージインターフェースを備え得る。例えば、ストレージインターフェースは、例えば、デバイスに含まれるメモリへのインターフェースとして、例えば、各デバイスのメモリ内にローカルに実装され得る。また、ストレージインターフェースはオフラインの、例えばローカルではないストレージ、例えばクラウドストレージ、例えば別のデバイス内に設けられたメモリまたはドライブなどのストレージとのインターフェースであってもよい。クラウドストレージが使用される場合、デバイスはさらにローカルストレージ（例えば、メモリ）を含み得る。例えば、メモリはコンピュータプログラミング命令や一時ファイルなどを保存するために使用され得る。

20

【0139】

デバイス300および350の様々な実施形態において、様々な選択肢から通信インターフェースが選択され得る。例えば、インターフェースは、ローカルまたはワイドエリアネットワーク、例えばインターネットへのネットワークインターフェースであってもよく、または内部もしくは外部データストレージへのストレージインターフェース、またはアプリケーションインターフェース（API）などであり得る。

【0140】

デバイス300および350は、1つまたは複数のボタン、キーボード、ディスプレイ、タッチスクリーンなどの周知の要素を含み得るユーザインターフェースを有し得る。ユーザインターフェースは、鍵共有プロトコルを開始するためのユーザインタラクションへの対応、鍵共有プロトコルへの応答、暗号化されたメッセージの送信等のために構成され得る。

30

【0141】

ストレージは、例えば、フラッシュメモリ等の電子メモリや、ハードディスク等の磁気メモリとして実装されてもよい。ストレージは、ストレージを構成する複数の個別のメモリを備えていてもよい。ストレージは一時的なメモリ、例えばRAMであってもよい。

【0142】

典型的には、デバイス300および350はそれぞれ、デバイス300および350に保存された適切なソフトウェアを実行するマイクロプロセッサを備える。例えば、ソフトウェアは、対応するメモリ、例えばRAMのような揮発性メモリ、またはフラッシュのような不揮発性メモリにダウンロードおよび／または保存されていてもよい。あるいは、デバイス300および350は、全体的にまたは部分的に、例えばフィールドプログラマブルゲートアレイ（FPGA）として、プログラマブルロジックに実装されてもよい。デバイス300および350は、全体的にまたは部分的に、いわゆる特定用途向け集積回路（ASIC）、例えば、特定の用途向けにカスタマイズされた集積回路（IC）として実装されてもよい。例えば、回路は、例えばVerilog、VHDLなどのハードウェア記述言語を使用して、CMOSで実装されてもよい。

40

【0143】

50

一実施形態では、デバイス300および350は、それぞれのデバイスの1つ以上の機能または全ての機能を実装するための1つまたは複数の回路を備え得る。これらの回路は、本明細書に記載される対応する機能を実装し得る。回路は、プロセッサ回路および記憶回路であってもよく、プロセッサ回路は、記憶回路において電子的に表現される命令を実行する。

【0144】

プロセッサ回路は分散された形で、例えば複数のサブプロセッサ回路として実装されてもよい。ストレージは、複数の分散サブストレージに分散されてもよい。メモリの一部または全部が電子メモリや磁気メモリ等であり得る。例えば、ストレージは、揮発性部分および不揮発性部分を有し得る。ストレージの一部が読み取り専用であってもよい。また、回路はFPGA、ASICなどであってもよい。

10

【0145】

第1のデバイス300は、第1の秘密鍵(r)および第1の秘密鍵から導出される第1の公開鍵(b)を生成するように構成された公開／秘密鍵生成器315を備え得る。秘密鍵から公開鍵を導出するために共通オブジェクト(a)が使用され得る。例えば、公開鍵を生成することは、共通オブジェクトと乗算すること、および／または何らかのノイズを導入すること(例えば、乗算結果のスケールダウン、ノイズ項の追加など)を含み得る。秘密鍵および共通オブジェクトは(例えば、有限体または環上の)多項式または行列であり得る。

【0146】

第1のデバイス300および第2のデバイス350は共通のオブジェクトaについて合意するように構成される。例えば、第1のデバイス300および第2のデバイス350は、第1のデバイス300および第2のデバイス350に／から送られる、最終シードの計算に使用され得るデータ(例えば、1つまたは複数のプレシード σ)を交換するように構成され得る。例えば、第1のデバイス300および第2のデバイス350は、同様に最終シードの計算に使用され得る公開データ、例えばカウンタやネットワークデータなどを維持するように構成され得る。例えば、第1のデバイス300および第2のデバイス350は、事前共有秘密pなどの秘密データを保存するように構成されてもよい。例えば、第1のデバイス300および第2のデバイス350は、例えば、乱数ビーコンから共通データを取得するように構成されてもよい。これらのソースの1つまたは複数について、各デバイスは共通の最終シード τ を計算する。攻撃者が同じ最終シード τ に到達するには、攻撃者は同じソースへのアクセスを有するか、または同じソースを追跡する必要がある。デバイス300およびデバイス350は共通の最終シードから共通オブジェクトaを導出するように構成される。なお、鍵共有に使用される共通オブジェクトは攻撃者に知られない。

20

30

【0147】

第1の秘密鍵および公開鍵は一時的に生成されてもよい。例えば、後者は、特に第1および第2のデバイスが別の認証メカニズム(例えばアウトオブバンドメカニズム、例えば証明書ベースの認証など)を使用して互いを認証する場合、鍵共有プロトコルのために行われ得る。

【0148】

第1の公開鍵は、例えば通信インターフェース305および355を介して、第1のデバイス300から第2のデバイス350に伝送される。これは直接通信によって行われ得る。必要に応じて、第1の公開鍵とともにプレシード σ が伝送されてもよい。例えば、プレシード σ を送ることによって、共通オブジェクト(a)の生成に使用され得る最終シードが計算され得る。

40

【0149】

第2のデバイス350は公開鍵取得器360を備え得る。公開鍵は第1のデバイスから直接取得されてもよい(場合によってはアウトオブバンドで、例えば電子メールで)。公開鍵は必要になるまで保存されていてもよい。しかし、第1の公開鍵はまた、即時使用のために、例えば鍵共有動作のために受け取られてもよく、例えば、この場合は第1の公

50

公開鍵および／または共通オブジェクトが一時的に生成されてもよい。

【0150】

第2のデバイス350は、第2の秘密鍵(s)を生成するように、および第2の秘密鍵(s)から第2の公開鍵(u)を生成するように構成された公開／秘密鍵生成器365を備え得る。第2の公開鍵は、第1の公開鍵の生成と同じ共通オブジェクトを使用する。第1および第2の秘密鍵はそれぞれのデバイスのみが知る秘密である。必要に応じて、例えばバックアップやキーエスクローなどのために、これらが信頼できるパーティ間で共有されてもよい。公開鍵および共通オブジェクトはセキュリティのために秘密でなければならないとは限らない。しかし、必要に応じて、1つ以上が第1および第2のデバイスのみが知る秘密であってもよい。例えば、第1の公開鍵は第2のデバイスとしか共有されなくてもよく、その逆も可能である。

10

【0151】

独立した生成は、(例えばメッセージの場合には)メッセージが、公開鍵暗号化から独立したアプリケーション(例えば、金融または通信アプリケーションなど)から生成される場合に得られ得る。独立した生成は、例えばランダム生成によって得られ得る。

【0152】

第2のデバイス350は生の鍵生成器375を備え得る。生の鍵生成器375は、第1の公開鍵(b)および第2の秘密鍵(s)から第2の生の共有鍵(k*)を生成するように構成される。例えば、生の鍵生成器375は、第1の公開鍵および第2の秘密鍵にノイズな乗算を適用するように構成され得る。例えば、ノイズな乗算は、基礎となるメカニズムに応じて乗算またはべき乗であり得る。第2のデバイス350は自身の第2の公開鍵を第1のデバイス300に伝送するように構成される。

20

【0153】

第1のデバイス300は生の鍵生成器325を備え得る。生の鍵生成器325は、例えばノイズな乗算を適用することによって、第2の公開鍵(u)および第1の秘密鍵(r)から第1の生の共有鍵(k')を生成するように構成される。残念ながら、ノイズな乗算では、第1および第2の生の鍵が必ずしも同一ではなく、互いに近い可能性がある。これが生じる具体的な可能性は基礎となるノイズな乗算によって異なる。ほとんどの用途では生の鍵が互いに異なる可能性がある程度受け入れられ得るが、許容される可能性の高さは用途に依存する。ただし、通常はこの可能性が低いことが好ましい。生の鍵は、秘密鍵および公開鍵と同じ数学的タイプのもの、例えば多項式または行列であってもよい。

30

【0154】

調整データ生成器385は、生の共有鍵の指し示された係数のための調整データ(h)を生成するように構成される。調整データは、第1および第2のデバイスにおいて導出された第1および第2の生の鍵の間の差異を低減することを可能にする情報を含む。例えば、調整データを適用すると、第1のデバイスと第2のデバイスの生の鍵の係数間の差(例えば、リー距離)が低減し、結果として、両者が同じビットを生成する可能性が高くなる。調整データは生の共有鍵内の係数上で計算されてもよい。

【0155】

調整データを実装する1つの方法は、鍵ビットとして選択されたビットに続く1つ以上の、例えば b_h 個の係数のビットを取得することである。例えば、これらは重要性においてB個のビットに続く b_h 個のビットであり得る。例えば、選択された係数ごとの調整ビットの数は、例えば1または2であり得る。調整ビットの数が少ないと、通信のオーバーヘッドが削減されるという利点がある。ただし、より大きな量の調整ビットも可能である。

40

【0156】

第2のデバイス350はカプセル化器390を含み得る。カプセル化器390はカプセル化関数、例えばXORを適用することによって、鍵ビットを用いて鍵Kをカプセル化するように構成される。カプセル化はワнтаイム・パッド・カプセル化であり得る。例えば、XOR関数、または本明細書に記載の他のカプセル化関数のうちの1つが使用され得る。

【0157】

50

第2のデバイスは、第2の公開鍵（ u ）、調整データ（ h ）、および／またはカプセル化されたデータ（ c ）を伝送するように構成される。伝送は第1の公開鍵の受け取りに応答して行われてもよいし（例えば、鍵共有の場合）、そうでなくてもよい（例えば、公開鍵暗号化の場合）。

【0158】

第1のデバイス300は、第2のデバイスから第2の公開鍵（ u ）、調整データ（ h ）、およびカプセル化されたデータ（ c ）を受け取るように構成される。第1のデバイス300は、第1の生の共有鍵（ k' ）の係数に調整関数内の調整データ（ h ）を適用することで鍵ビット（ k ）を取得するように構成された調整ユニット335を備え得る。例えば、係数は調整ビットを使用して調整され、その後、高信頼性ビットを得るためにサンプリングされ得る。第1のデバイス300は、カプセル化されたデータ（ c ）をカプセル化解除して鍵 K を取得するように構成されたカプセル化解除ユニット340を備える。一実施形態では、デバイス300およびデバイス350は、調整ユニットおよびカプセル化／カプセル化解除ユニットのうち的一方のみを含む。一実施形態では、デバイス300およびデバイス350は、それぞれの生の鍵の間の距離を短くする他の方法を含む。

【0159】

図5aは、一実施形態に係る暗号方法400の例を概略的に示す。例えば、方法400は方法450と一緒に使用されてもよい。例えば、方法400および方法450は、本明細書に記載の実施形態に従って拡張または変更されてもよい。第2の暗号方法（400）は、

第1の暗号デバイス（10）と通信するステップ（405）と、

第1の暗号デバイス（10）から受け取られたプレシード、および

第2の暗号デバイスと第1の暗号デバイスとの間で事前に共有される事前共有秘密から最終シードを計算するステップ（410）と、

最終シードから共通オブジェクト（ a ）を計算するステップ（415）と、

第1の暗号デバイスに関連付けられた第1の公開鍵（ b ）を取得し、第2の暗号デバイスに関連付けられた第2の秘密鍵（ s ）を取得するステップ（420）と、

第2の秘密鍵（ s ）から第2の公開鍵（ u ）を計算するステップであって、第2の公開鍵（ u ）の計算は、第2の秘密鍵（ s ）に共通オブジェクト（ a ）を乗算することを含む、ステップ（425）と、

第1の公開鍵（ b ）および第2の秘密鍵（ s ）から第2の生の共有鍵（ k^* ）を計算するステップであって、第2の生の共有鍵の計算は、第2の秘密鍵（ s ）と第1の公開鍵（ b ）との間の乗算を含む、ステップ（430）と、

第2の公開鍵（ u ）を第1のデバイスに伝送するステップ（435）とを含み得る。

【0160】

図5bは、一実施形態に係る暗号方法450の例を概略的に示す。第1の暗号方法（450）は、

第2の暗号デバイス（20）と通信するステップ（455）と、

プレシードを選択し、プレシードを第2の暗号デバイス（20）に送るステップ（460）と、

プレシード、および第2の暗号デバイスと第1の暗号デバイスとの間で事前に共有される事前共有秘密から最終シードを計算するステップ（465）と、

最終シードから共通オブジェクト（ a ）を計算するステップ（470）と、

第1の暗号デバイスに関連付けられた第1の秘密鍵（ r ）を取得し、第1の秘密鍵（ r ）を共通オブジェクト（ a ）と乗算することを含む計算により、第1の秘密鍵（ r ）から第1の公開鍵（ b ）を計算し、第1の公開鍵（ b ）を第2の暗号デバイスに伝送するステップ（475）と、

第2の暗号デバイスから第2の公開鍵（ u ）を受け取るステップ（480）と、

第2の公開鍵（ u ）および第1の秘密鍵（ r ）から第1の生の共有鍵（ k' ）を計算するステップであって、第1の生の共有鍵の計算は、第2の公開鍵（ u ）と第1の秘密鍵（ r ）

10

20

30

40

50

）との間の乗算を含む、ステップ（４８５）とを含み得る。

【０１６１】

例えば、方法４００および４５０はコンピュータにより実装された方法であってもよい。例えば、方法の計算部分はコンピュータや計算回路などを用いて計算されてもよい。例えば、暗号デバイスとの通信は通信インターフェースを使用して行われてもよい。例えば、パラメータや鍵などの格納または取り出しは電子ストレージ、例えばメモリ、ハードドライブ等から行われ得る。鍵および／または共通オブジェクトは、１０バイト、５０バイト、１００バイト等より大きいデータで表されてもよい。

【０１６２】

当業者には明らかであるように、方法を実行する多くの異なる態様が存在する。例えば、ステップは記載または図示されている順序で実行されてもよいが、ステップの順序を変更したり、一部のステップを並行して実行したりすることも可能である。さらに、ステップ間に他の方法ステップを挿入することができる。挿入されるステップは、本明細書に記載されるような方法の改良を表し、または方法と無関係であってもよい。さらに、あるステップは、次のステップが開始される前に完全に終了していなくてもよい。

【０１６３】

方法の実施形態は、プロセッサシステムに方法４００および／または４５０を実行させるための命令を含むソフトウェアを使用して実行され得る。ソフトウェアは、システムの特定のサブエンティティによって取られるステップのみを含んでもよい。ソフトウェアは、ハードディスク、フロッピー、メモリ、光ディスクなどの適切な記憶媒体に保存されてもよい。ソフトウェアは、有線または無線を介する信号として、またはインターネットなどのデータネットワークを使用して送られてもよい。ソフトウェアは、サーバ上でダウンロードおよび／またはリモート使用可能であってもよい。方法の実施形態は、方法を実行するためにプログラマブルロジック、例えばフィールドプログラマブルゲートアレイ（ＦＰＧＡ）を構成するように構成されたビットストリームを使用して実行されてもよい。

【０１６４】

本開示の主題は、コンピュータプログラム、特に、本開示の主題を実施するように適合された、キャリア上のまたはキャリア内のコンピュータプログラムにも及ぶことが理解されよう。プログラムは、ソースコード、オブジェクトコード、コード中間ソース、部分的にコンパイルされた形式などのオブジェクトコード、または方法の実施形態の実装のための使用に適した任意の他の形式などの形式を取り得る。コンピュータプログラム製品に関する実施形態は、上記方法のうちの少なくとも１つの方法の処理ステップに対応するコンピュータ実行可能命令を含む。これらの命令は、サブルーチンに細分され、かつ／または静的にもしくは動的にリンクされ得る１つまたは複数のファイルに保存され得る。コンピュータプログラム製品に関する他の実施形態は、少なくとも１つの上記システムおよび／または製品のデバイス、ユニット、および／または部分に対応するコンピュータ実行可能命令を含む。

【０１６５】

図６ａは、一実施形態に係る、コンピュータプログラム１０２０を含む書き込み可能部分１０１０を有するコンピュータ可読媒体１０００を示し、コンピュータプログラム１０２０は、プロセッサシステムに暗号方法を実行させるための命令を含む。コンピュータプログラム１０２０は、物理的なマークとして、またはコンピュータ可読媒体１０００の磁化によって、コンピュータ可読媒体１０００上に具現化され得る。しかしながら、任意の他の適切な実施形態もまた考えられる。さらに、ここではコンピュータ可読媒体１０００が光ディスクとして示されているが、コンピュータ可読媒体１０００は、ハードディスク、ソリッドステートメモリ、フラッシュメモリなどの任意の適切なコンピュータ可読媒体であってもよく、また、記録不能または記録可能であってもよい。コンピュータプログラム１０２０は、プロセッサシステムに前記暗号方法を実行させるための命令を含む。

【０１６６】

図６ｂは、暗号デバイスの一実施形態に係るプロセッサシステム１１４０の概略図であ

10

20

30

40

50

る。プロセッサシステムは1つまたは複数の集積回路1110を備える。1つまたは複数の集積回路1110のアーキテクチャが図6bに概略的に示されている。回路1110は、一実施形態に係る方法を実行するために、かつ／または、モジュールもしくはユニットを実装するためにコンピュータプログラムコンポーネントを実行するための処理ユニット1120（例えばCPUなど）を備える。回路1110はプログラミングコード、データ等を保存するためのメモリ1122を含む。メモリ1122の一部は読み出し専用であってもよい。回路1110は通信要素1126、例えばアンテナもしくはコネクタ、または両方を備え得る。回路1110は、方法において定義される処理の一部または全部を実行するための専用集積回路1124を備え得る。プロセッサ1120、メモリ1122、専用IC1124、および通信要素1126は相互接続1130、例えばバスを介して相互に接続され得る。プロセッサシステム1110は、アンテナおよび／またはコネクタを用いた接触および／または非接触通信のために構成され得る。

【0167】

例えば、一実施形態では、プロセッサシステム1140（例えば、暗号デバイス）はプロセッサ回路およびメモリ回路を備え、プロセッサは、メモリ回路内に格納されたソフトウェアを実行するように構成される。例えば、プロセッサ回路はIntel Core i7プロセッサやARM Cortex-R8等であり得る。一実施形態では、プロセッサ回路はARM Cortex-M0であり得る。メモリ回路はROM回路、または不揮発性メモリ、例えばフラッシュメモリであってもよい。メモリ回路は揮発性メモリ、例えばSRAMメモリであってもよい。後者の場合、デバイスは、ソフトウェアを提供するように構成された不揮発性ソフトウェアインターフェース（例えば、ハードドライブ）やネットワークインターフェース等を備え得る。

【0168】

デバイス1140は、記載された各コンポーネントを1つずつを含むものとして示されているが、様々なコンポーネントは様々な実施形態において複製されてもよい。例えば、処理ユニット1120（例えば、プロセッサ1120）は、本明細書に記載の方法を独立して実行するように構成された複数のマイクロプロセッサを含んでもよく、または、本明細書に記載の方法のステップもしくはサブルーチンを複数のプロセッサが協働して実行することで本明細書に記載の機能を達成するように構成された複数のマイクロプロセッサを含んでもよい。さらに、デバイス1140がクラウドコンピューティングシステム内に実装される場合、様々なハードウェアコンポーネントが別々の物理システムに属し得る。例えば、プロセッサ1120は第1のサーバ内に第1のプロセッサを含み、第2のサーバ内に第2のプロセッサを含み得る。

【0169】

上記実施形態は本開示の主題を限定ではなく説明するものであり、当業者は多くの代替的な実施形態を設計できることに留意されたい。

【0170】

特許請求の範囲において、括弧間の参照符号は請求項を限定するものとして解釈されるべきではない。「含む」という動詞およびその活用形の使用は、請求項に記載された要素またはステップ以外の要素またはステップの存在を排除するものではない。要素の単数形は、複数のかかる要素の存在を排除するものではない。列挙された要素に付随する「～のうちの少なくとも1つ」などの表現は、列挙された要素のうちの全ての要素または任意の部分集合の選択を表す。例えば、「A、B、およびCのうちの少なくとも1つ」という表現は、Aのみ、Bのみ、Cのみ、AおよびBの両方、AおよびCの両方、BおよびCの両方、またはA、B、C全てを含むものと理解されたい。本開示の主題は、複数の別個の要素を含むハードウェアによって、および適切にプログラムされたコンピュータによって実装され得る。いくつかの部分の部分を列挙する装置クレームにおいて、これらの部分のうちのいくつかは、同一のハードウェアアイテムによって具現化されてもよい。複数の手段が互いに異なる従属請求項に記載されているからといって、これらの手段の組み合わせを好適に使用することができないとは限らない。

10

20

30

40

50

【0171】

以下の番号付きの各項には考えられる非限定的な例が含まれる。

【0172】

第1項

第2の暗号デバイス(20)であって、前記第2の暗号デバイスは、

第1の暗号化デバイス(10)と通信するように構成された通信インターフェースと、
プロセッサとを備え、前記プロセッサは、

第1の暗号デバイス(10)から受け取られたプレシード、および

第2の暗号デバイスと第1の暗号デバイスとの間で事前に共有される事前共有秘密
から最終シードを計算し、

最終シードから共通オブジェクト(a)を計算し、

第1の暗号デバイスに関連付けられた第1の公開鍵(b)を取得し、第2の暗号デバ
イスに関連付けられた第2の秘密鍵(s)を取得し、

前記第2の秘密鍵(s)から第2の公開鍵(u)を計算し、ここで、前記第2の公開
鍵(u)を計算することは前記第2の秘密鍵(s)を前記共通オブジェクト(a)と乗算
することを含み、

前記第1の公開鍵および前記第2の秘密鍵から第2の生の共有鍵を計算し、ここで、
前記第2の生の共有鍵の計算は、前記第2の秘密鍵と前記第1の公開鍵との間の乗算を含
み、

前記第2の公開鍵(u)を前記第1のデバイスに伝送する、第2の暗号デバイス。

【0173】

第2項

第1の暗号デバイスで(10)あって、前記第1の暗号デバイスは、

第2の暗号デバイス(20)と通信する通信インターフェースと、

プロセッサとを備え、前記プロセッサは、

プレシードを選択し、前記プレシードを前記第2の暗号デバイス(20)に送り、

前記プレシード、および前記第2の暗号デバイスと前記第1の暗号デバイスとの間で
事前に共有される事前共有秘密から最終シードを計算し、

最終シードから共通オブジェクト(a)を計算し、

前記第1の暗号デバイスに関連付けられた第1の秘密鍵(r)を取得し、前記第1の
秘密鍵(r)を前記共通オブジェクト(a)と乗算することを含む計算により、前記第1
の秘密鍵(r)から第1の公開鍵(b)を計算し、前記第1の公開鍵(b)を前記第2の
暗号デバイスに伝送し、

前記第2の暗号デバイスから第2の公開鍵(u)を受け取り、

前記第2の公開鍵(u)および前記第1の秘密鍵(r)から第1の生の共有鍵(k')
を計算し、ここで、前記第1の生の共有鍵の計算は、前記第2の公開鍵(u)と前記第1
の秘密鍵(r)との間の乗算を含む、第1の暗号デバイス。

【0174】

第3項

前記事前共有秘密は事前共有パスワードを含む、第1項または第2項に記載の第2の暗
号デバイス(20)または第1の暗号デバイス(10)。

【0175】

第4項

前記プレシードは平文形式で送受信される、第1項から第3項のいずれか一項に記載の
第2の暗号デバイス(20)または第1の暗号デバイス(10)。

【0176】

第5項

前記プレシードは前記通信インターフェースを介して送受信されるが、前記事前共有秘
密は前記通信インターフェースを介して送受信されない、第1項から第4項のいずれか一
項に記載の第2の暗号デバイス(20)または第1の暗号デバイス(10)。

10

20

30

40

50

【0177】

第6項

前記最終シードはさらに、前記第2の暗号デバイスと前記第1の暗号デバイスとの間のネットワーク接続からのネットワーク構成データから計算される、第1項から第5項のいずれか一項に記載の第2の暗号デバイス(20)または第1の暗号デバイス(10)。

【0178】

第7項

前記最終シードは、

前記第1および／または第2の暗号デバイスのネットワークアドレス、

ネットワーク接続の目的、

時刻、

インタラクションカウンタから計算され、前記インタラクションカウンタは、前記第2の暗号デバイス(20)および前記第1の暗号デバイス(10)によって維持され、少なくとも、共有鍵が導出されるたびに増加される、第1項から第6項のいずれか一項に記載の第2の暗号デバイス(20)または第1の暗号デバイス(10)。

【0179】

第8項

前記第2の暗号デバイス(20)の前記プロセッサは、

送信鍵(K)を生成し、

カプセル化関数を適用することによって、前記第2の生の共有鍵の少なくとも一部を用いて前記送信鍵をカプセル化して、カプセル化されたデータ(c)を取得し、

前記カプセル化されたデータ(c)を前記第1の暗号デバイスに送り、

前記第1の暗号デバイス(10)の前記プロセッサは、

前記第2のデバイスから前記カプセル化されたデータ(c)を受け取り、

前記第1の生の共有鍵(k')の少なくとも一部を使用して、前記カプセル化されたデータ(c)のカプセル化を解除することで送信鍵を取得する、第1項から第7項のいずれか一項に記載の第2の暗号デバイス(20)または第1の暗号デバイス(10)。

【0180】

第9項

前記第2の暗号デバイス(20)の前記プロセッサは、

前記生の共有鍵の係数の少なくとも一部のための調整データ(h)を生成し、ここで、前記調整データは、前記第1および第2のデバイスにおいて導出された前記第1および第2の生の鍵の間の差を低減することを可能にする情報を含み、

前記第2の生の共有鍵(k*)から送信鍵(K)を生成し、

前記調整データ(h)を前記第1の暗号デバイスに送り、

前記第1の暗号デバイス(10)の前記プロセッサは、

前記第2のデバイスから調整データ(h)を受け取り、

前記第1の生の共有鍵(k')内の係数の少なくとも一部に対して、調整関数において前記調整データ(h)を適用し、

前記調整された第1の生の共有鍵(k')から送信鍵(K)を生成する、第1項から第8項のいずれか一項に記載の第2の暗号デバイス(20)または第1の暗号デバイス(10)。

【0181】

第10項

前記第1の公開鍵(b)および前記第2の公開鍵(u)を計算する際の前記共通オブジェクト(a)との前記乗算はノイジーな乗算である、第1項から第9項のいずれか一項に記載の第2の暗号デバイス(20)または第1の暗号デバイス(10)。

【0182】

第11項

前記第2の暗号デバイスの前記プロセッサは、前記送信鍵を用いてメッセージ(m)を

10

20

30

40

50

暗号化して、前記暗号化されたメッセージを前記第1の暗号デバイスに送り、前記第1の暗号デバイスの前記プロセッサは、前記送信鍵を用いて前記暗号化されたメッセージを復号して、前記メッセージ(m)を検証し、かつ／または

前記第1の暗号デバイスの前記プロセッサは、前記送信鍵を使用してメッセージ(m)を暗号化して、前記暗号化されたメッセージを前記第2の暗号デバイスに送り、前記第2の暗号デバイスの前記プロセッサは、前記送信鍵を用いて前記暗号化されたメッセージを復号して、前記メッセージ(m)を検証する、第1項から第10項のいずれか一項に記載の第2の暗号デバイスまたは第1の暗号デバイス。

【0183】

第12項

前記送信鍵はランダムであり、かつ／または一時的であり、かつ／または対称であり、かつ／または前記第1の公開鍵(b)から独立している、第1項から第11項のいずれか一項に記載の第2の暗号デバイスまたは第1の暗号デバイス。

【0184】

第13項

前記第1および第2の公開鍵、前記第1および第2の秘密鍵、前記第1および第2の生の鍵、および前記共通オブジェクトは有限体または環上の行列であり、

前記第1および第2の公開鍵、前記第1および第2の秘密鍵、前記第1および第2の生の鍵、および前記共通オブジェクトは有限体または環上の多項式である、第1項から第12項のいずれか一項に記載の第2の暗号デバイスまたは第1の暗号デバイス。

【0185】

第14項

ネットワーク構成データは、前記送信鍵を用いて暗号化されて送られる、第1項から第13項のいずれか一項に記載の第2の暗号デバイスまたは第1の暗号デバイス。

【0186】

第15項

第2の暗号方法(400)であって、前記第2の暗号方法は、

第1の暗号デバイス(10)と通信するステップ(405)と、

第1の暗号デバイス(10)から受け取られたプレシード、および

第2の暗号デバイスと第1の暗号デバイスとの間で事前に共有される事前共有秘密から最終シードを計算するステップ(410)と、

最終シードから共通オブジェクト(a)を計算するステップ(415)と、

第1の暗号デバイスに関連付けられた第1の公開鍵(b)を取得し、第2の暗号デバイスに関連付けられた第2の秘密鍵(s)を取得するステップ(420)と、

第2の秘密鍵(s)から第2の公開鍵(u)を計算するステップであって、第2の公開鍵(u)の計算は、第2の秘密鍵(s)に共通オブジェクト(a)を乗算することを含む、ステップ(425)と、

前記第1の公開鍵(b)および前記第2の秘密鍵(s)から第2の生の共有鍵(k*)を計算するステップであって、前記第2の生の共有鍵を計算することは、前記第2の秘密鍵(s)と前記第1の公開鍵(b)との間の乗算を含む、ステップ(430)と、

第2の公開鍵(u)を第1のデバイスに伝送するステップ(435)とを含む、第2の暗号方法。

【0187】

第16項

第1の暗号方法(450)であって、前記第1の暗号方法は、

第2の暗号デバイス(20)と通信するステップ(455)と、

プレシードを選択し、プレシードを第2の暗号デバイス(20)に送るステップ(460)と、

前記プレシード、および前記第2の暗号デバイスと前記第1の暗号デバイスとの間で事前に共有される事前共有秘密から最終シードを計算するステップ(465)と、

10

20

30

40

50

最終シードから共通オブジェクト (a) を計算するステップ (470) と、

第1の暗号デバイスに関連付けられた第1の秘密鍵 (r) を取得し、第1の秘密鍵 (r) を共通オブジェクト (a) と乗算することを含む計算により、第1の秘密鍵 (r) から第1の公開鍵 (b) を計算し、第1の公開鍵 (b) を第2の暗号デバイスに伝送するステップ (475) と、

前記第2の暗号デバイスから第2の公開鍵 (u) を受け取るステップ (480) と、

第2の公開鍵 (u) および第1の秘密鍵 (r) から第1の生の共有鍵 (k') を計算するステップであって、第1の生の共有鍵の計算は、第2の公開鍵 (u) と第1の秘密鍵 (r) との間の乗算を含む、ステップ (485) とを含む、第1の暗号方法。

【0188】

第17項

プロセッサシステムによって実行されると、第15項および／または第16項に記載の方法を前記プロセッサシステムに実行させる命令を表すデータ (1020) を含む、一時的または非一時的なコンピュータ可読媒体 (1000)。

【0189】

特許請求の範囲において、括弧内の参照符号は、例示的实施形態の図面内の参照符号または実施形態の数式を指し、したがって、請求項の明瞭さを高める。これらの参照符号は請求項を限定するものとして解釈されるべきではない。

10

20

30

40

50

【図面】
【図 1】

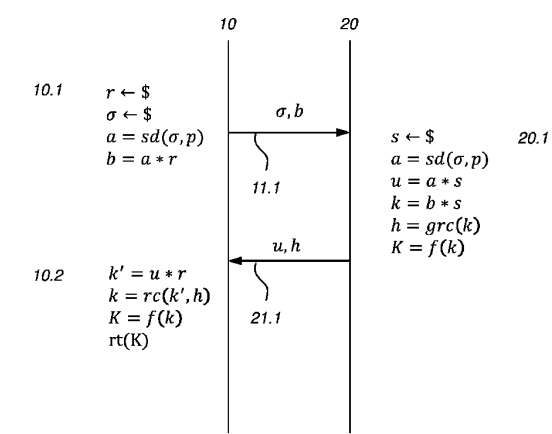


Fig. 1

【図 2】

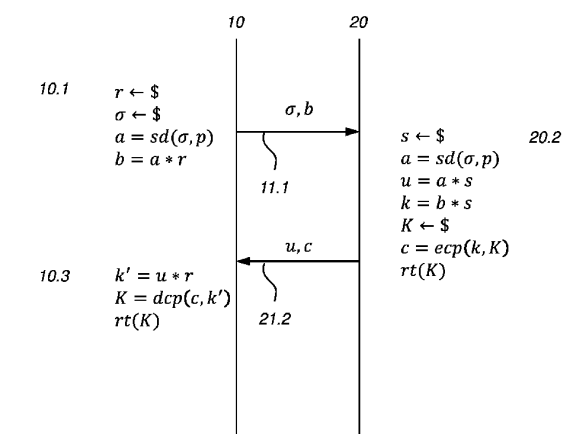


Fig. 2

【図 3】

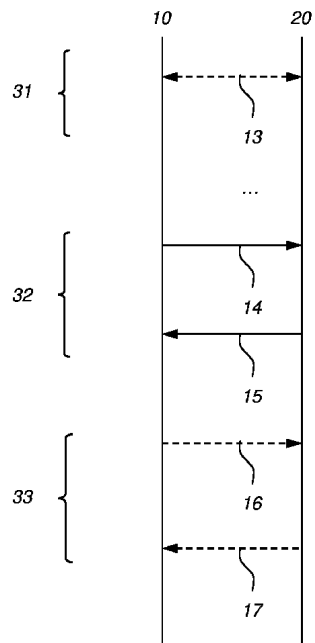


Fig. 3

【図 4】

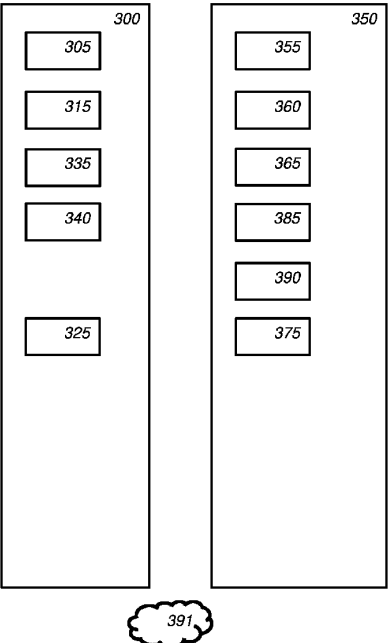


Fig. 4

10

20

30

40

50

【図 5 a】

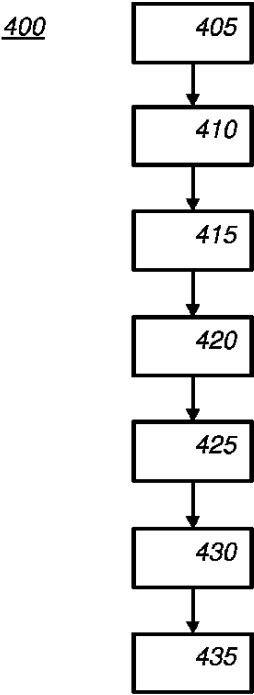


Fig. 5a

【図 5 b】

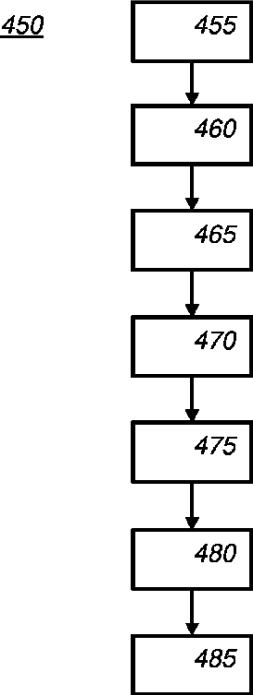


Fig. 5b

【図 6 a】

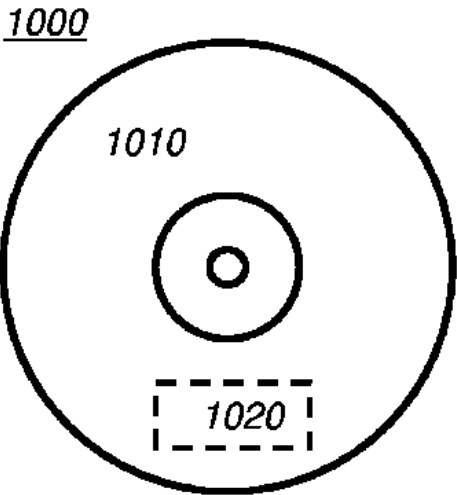


Fig. 6a

【図 6 b】

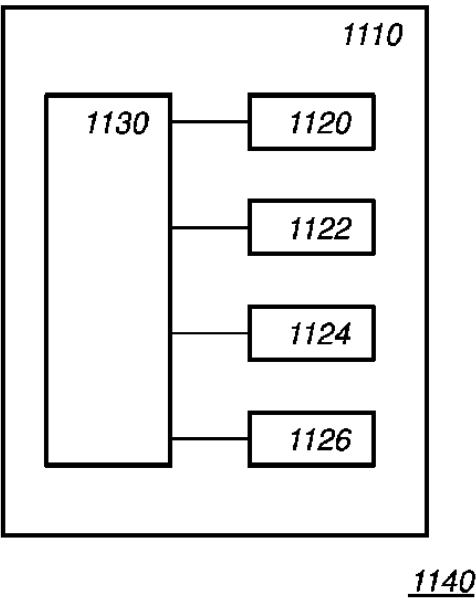


Fig. 6b

10

20

30

40

50

フロントページの続き

- (72)発明者 トルフィツェン ルドヴィクス マリヌス ジェラルダス マリア
オランダ国 5 6 5 6 アーエー アインドーフェン ハイ テック キャンパス 5
- (72)発明者 バタチャリア サウヴィク
オランダ国 5 6 5 6 アーエー アインドーフェン ハイ テック キャンパス 5

審査官 青木 重徳

- (56)参考文献 国際公開第2019/076737 (WO, A1)
Xinwei Gao et al., Efficient Implementation of Password-Based Authenticated Key Exchange from RLWE and Post-Quantum TLS, Cryptology ePrint Archive, Paper 2017/1192, [オンライン], 2017年12月18日, pp. 1-8, (2024年5月20日 検索)、インターネット, 〈URL: <https://eprint.iacr.org/2017/1192.pdf>〉
Ye Yuan et al., Protable implementation of post-quantum encryption schemes and key exchange protocols on JavaScript-enabled platforms, 2018年 暗号と情報セキュリティシンポジウム (SCIS2018) 予稿集 [USB] 2018年 暗号と情報セキュリティシンポジウム概要集 Abstracts of 2018 Symposium on Cryptography and Information Security, 日本, 2018年01月23日, 3A4-4, pp. 1-8
古川 智也 ほか, 通信データの特徴を用いた Ring-LWE 問題に基づく鍵交換プロトコルの通信量削減方法の検討, CSS2016 コンピュータセキュリティシンポジウム2016 論文集 [CD-ROM], 日本, 一般社団法人 情報処理学会, 2016年10月04日, Vol. 2016 No. 2, pp. 977-980
- (58)調査した分野 (Int.Cl., DB名)
H04L 9/08
H04L 9/32
JSTPlus/JMEDPlus/JST7580 (JDreamIII)