TTC標準
Standard

JT-Y3803

量子鍵配送ネットワーク - 鍵管理

Quantum key distribution networks - Key management

第 2.0 版

2024 年 2 月 15 日制定

一般社団法人
情報通信技術委員会

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE

TTC Telecommunication Technology Committee

目　次

＜参考＞ 4
要約 5
1.　規定範囲 5
2.　参照文献 5
3.　定義 5
3.1.　本標準以外で定義された用語 6
3.2.　本標準で定義する用語 7
4.　略語 7
5.　表記法 7
6.　鍵管理の概要 7
7.　鍵管理の機能要素 9
7.1.　KM 内のエージェント 10
7.2.　KM リンク 11
7.3.　参照点 11
7.4.　セキュリティ分界点 11
8.　鍵管理の詳細手順 11
8.1.　量子レイヤにおける QKD-鍵の生成 12
8.2.　鍵管理レイヤにおける KMA-鍵と KSA-鍵の管理 13
8.2.1.　KMA における鍵取得、認証、格納 13
8.2.2.　暗号アプリケーションからの鍵要求の受信 14
8.2.3.　KMA 間の鍵リレー 14
8.2.4.　KSA から暗号アプリケーションへの鍵供給 15
8.2.5.　KMA 間の鍵リレーの再ルーティング 16
8.2.6.　鍵ライフサイクル管理 17
9.　鍵リレーのいくつかのスキーム 17
10.　鍵ファイルフォーマット 19
参考文献 23

＜参考＞

1. 国際勧告などとの関連

本標準は量子鍵配送ネットワークの鍵管理について規定しており、2020 年 12 月に ITU-T SG13 において発行された ITU-T 勧告 Y.3803 に準拠している。

2. 上記勧告などに対する追加項目など

2.1 オプション選択項目

なし

2.2 ナショナルマター決定項目

なし

2.3 その他

なし

2.4 原勧告との章立て構成比較表

章立てに変更なし

3. 改版の履歴

版数	発行日	改版内容
第 1 版	2021 年 5 月 20 日	制定
第 1.1 版	2021 年 6 月 11 日	図中表記の和訳化、誤記訂正
第 2.0 版	2024 年 2 月 15 日	Amendment 1 の反映、誤記訂正

4. 工業所有権

本標準に関わる「工業所有権の実施の権利に係る確認書」の提出状況は、TTC ホームページでご覧になれます。

5. その他

(1) 参照している勧告、標準など

TTC 標準	JT-Y3800, JT-Y3801, JT-Y3802, JT-Y3804
ITU-T 勧告	X.1714

6. 標準作成部門

Network Vision 専門委員会

要約

量子鍵配送(QKD)プロトコルは、対称なランダムビット列を、無制限の計算能力を持つ盗聴者に対しても安全性を証明できる安全な鍵として配送する手段を提供する。QKD の基本要素は、QKD リンクによって接続された QKD モジュールのペアであり、これにより 2 つの離れた当事者が安全な鍵を共有できる。QKD ネットワーク(QKDN)は、2 以上の QKD リンクとトラステッドノード(QKD ノード)から構成されており、QKD リンクと鍵リレーを介して、任意の対となる 2 つの QKD ノード間で安全な鍵を共有することができる。最終的に、これらの鍵はユーザネットワークの暗号アプリケーションに提供される。QKDN を実装しユーザネットワークと適切に統合するために、ネットワーク能力、概念的構成、階層化モデル、基本機能とコンポーネント、およびユーザネットワークとの関係といった QKDN 技術の概要は、ITU-T 勧告 Y.3800 が規定する。

QKDN を効率的かつ安全に運用するためには、鍵管理が最も重要な課題となる。なぜならば、これ無くしては意義のある QKD の動作とサービスのほとんどが実現できない。

鍵管理には、少なくとも、QKD モジュールによって生成された鍵を格納すること、QKDN のノード間で鍵をリレーすること、およびユーザからの要求により暗号アプリケーションに鍵を供給すること、のすべてを安全な方法で行うことが含まれる。これらの課題の標準化は、QKDN の相互運用性を実現し、セキュリティを確保し、QKD のアプリケーションを拡大するために不可欠である。

本標準の目的は、QKDN の鍵管理の設計、導入、および運用のための支援を提供することである。QKDN の全体構造と基本機能は ITU-T 勧告 Y.3800 を参照し、QKDN の要求条件は ITU-T 勧告 Y.3801 を参照し、次に鍵管理の要求条件、機能要素及び手順を本標準で記述する。

1. 規定範囲

本標準は、量子鍵配送(QKD)ネットワークの鍵管理について記述し、実装と運用を支援するための技術仕様に対応する。この標準が規定する範囲は次のとおりである:

- QKDN の鍵管理の概要
- 鍵管理の機能要素
- 鍵管理の手順
- 鍵のフォーマット(鍵データとメタデータ)

2. 参照文献

以下に列挙する ITU-T 勧告およびその他の参照文献は、この本文中の参照を通して、本標準を構成する規定を含む。発行時点では、示された版は有効であった。すべての勧告及び他の参照文献は改訂の対象である。したがって、本標準の利用者は、以下に列挙する勧告及び他の参照文献の最新版を適用する可能性を調査することが推奨される。現在有効な ITU-T 勧告のリストは定期的に発行されている。本標準が文献を参照することは、その文献がそれ単体で勧告となる地位をその文献に与えるものではない。

[ITU-T X.1714] ITU-T X.1714(2020)、量子鍵配送ネットワークの鍵合成と秘密鍵の供給

[ITU-T Y.3800] ITU-T Y.3800(2019)、量子鍵配送ネットワークの概要

[ITU-T Y.3801] ITU-T Y.3801(2020)、量子鍵配送ネットワークの機能要求条件

[ITU-T Y.3802] ITU-T Y.3802(2020)、量子鍵配送ネットワークのアーキテクチャ

[ITU-T Y.3804] ITU-T Y.3801(2020)、量子鍵配送ネットワークの制御と管理

3. 定義

3.1. 本標準以外で定義された用語

本標準では、本標準以外で定義された次の用語を使用する。

3.1.1. 暗号ハッシュ関数[b-ETSI GR QKD007]：任意の長さのバイナリ列を固定長のバイナリ列にマッピングする計算効率の高い関数で、逆方向のマッピングを計算すること、および同一の値にマッピングされる２つの異なる値を見つけることが計算量的にできない。

3.1.2. ハッシュ値[b-ETSI GS QKD008]:暗号ハッシュ関数の出力。

3.1.3. 鍵ライフサイクル[ITU-T Y.3800]: 鍵マネージャ (KM) の鍵受信から、暗号アプリケーションでの鍵利用と鍵管理ポリシーによる削除または保存までの一連の処理。

3.1.4. 鍵管理[ITU-T Y.3800]: 量子レイヤからの受信、蓄積、フォーマッティング、リレー、同期、認証、暗号アプリケーションへの供給、鍵管理ポリシーによる削除または保存まで、鍵ライフサイクルで実行されるすべての動作。

3.1.5. 鍵管理エージェント(KMA)[ITU-T Y.3802]: QKD ノード（トラステッドノード）内の 1 つまたは複数の QKD モジュールによって生成された鍵を管理するための機能要素。

注 – KMA は、1 つまたは複数の QKD モジュールから鍵を取得し、同期、サイズ変更、フォーマット、および格納を行う。また、鍵管理エージェント(KMA)リンクを介して鍵のリレーを行う。

3.1.6. 鍵管理エージェントリンク[ITU-T Y.3802]：鍵管理エージェント(KMA)を接続する通信リンクで、IT セキュア鍵リレーと鍵管理のための通信を実行する。

3.1.7. 鍵マネージャ (KM) [ITU-T Y.3800]：鍵管理レイヤ内で鍵管理を実行する機能モジュールで、QKD ノード内に配置される。

3.1.8. 鍵マネージャ(KM)リンク[ITU-T Y.3800]：鍵マネージャ(KM) を接続し、鍵管理を行う通信リンク。

3.1.9. 鍵供給エージェント(KSA)[ITU-T Y.3802]：鍵管理エージェント(KMA)と暗号アプリケーションの間に位置し、暗号アプリケーションに鍵を供給する機能モジュール。

注 - 暗号アプリケーション用のアプリケーションインタフェースは、KSA に実装される。KSA は鍵を同期し、暗号アプリケーションに鍵を供給する前に KSA リンクを介してその完全性を検証する。

3.1.10. 鍵管理エージェント(KMA)リンク：鍵管理エージェント(KMA)を接続して鍵リレーと鍵管理のための通信の実行する通信リンク。

3.1.11. 量子鍵配送(QKD)[b-ETSI GR QKD007]:量子情報理論に基づく情報理論的セキュリティを用いて対称暗号鍵を生成および配送する手順または方法。

3.1.12. QKD-鍵[ITU-T Y.3802]: 1 対の QKD モジュールによって生成される一対の対称ランダムビット列。特に、KM でサイズ変更およびフォーマットされる前のランダムビット列を指す。

3.1.13. QKD リンク[ITU-T Y.3800]: QKD を動作させるための 2 つの QKD モジュール間の通信リンク。

注：QKD リンクは、量子信号を送受信する量子チャネルと、同期と鍵蒸留のために情報を交換する古典チャネルから構成される。

3.1.14. QKD モジュール[ITU-T Y.3800]：暗号機能と、QKD プロトコル、同期、鍵生成のための蒸留などの量子光プロセスを実装するハードウェアおよびソフトウェアコンポーネントのセット。定められた暗号境界内に含まれる。

注： QKD モジュールは、QKD リンクに接続され、鍵を生成するエンドポイントモジュールとして動作する。QKD モジュールには 2 つのタイプ、すなわち送信器 (QKD-Tx) および受信器 (QKD- Rx) がある。

3.1.15. QKD ネットワーク (QKDN) [ITU-T Y.3800]： QKD リンクを介して接続された 2 以上の QKD ノードから構成するネットワーク。

注： QKD ネットワーク (QKDN) では、QKD リンクで直接接続されていない QKD ノード間でも、鍵リレーによって鍵を共有できる。

3.1.16. QKDN コントローラ[ITU-T Y.3800]：QKDN を制御するために QKDN 制御レイヤに位置する機能モジュール。

3.1.17. QKDN マネージャ[ITU-T Y.3800]：QKDN を監視および管理するために QKDN 管理レイヤに位置する機能モジュール。

3.1.18. QKD ノード[ITU-T Y.3800]：許可されていない当事者による侵入および攻撃から保護されている 1 つ以上の QKD モジュールを含むノード。

注：QKD ノードは、鍵マネージャ(KM)を含むことができる。

3.1.19. ユーザネットワーク[ITU-T Y.3800]： QKDN によって供給される鍵を暗号アプリケーションが利用するネットワーク。

3.2. 本標準で定義する用語

本標準では、次の用語を定義する。

3.2.1. 鍵データ：ランダムビット列。暗号鍵として使用される。

3.2.2. 鍵管理エージェント鍵 (KMA-鍵)：鍵管理エージェント(KMA)で格納され処理される鍵データ。任意の KMA と組みとなる KMA の間で安全に共有される。

3.2.3. 鍵供給エージェント鍵 (KSA-鍵)：鍵供給エージェント(KSA)で格納され処理される鍵データ。任意の KSA と組みとなる KSA の間で安全に共有される。

4. 略語

本標準では、以下の略語を使用する：

ID	識別子（Identifier）
IT-secure	IT セキュア（Information-theoretically secure）
KM	鍵マネージャ（Key manager）
KMA	鍵管理エージェント（Key management agent）
KSA	鍵供給エージェント（Key supply agent）
OTP	ワンタイムパッド（One-Time Pad）
QBER	量子ビットエラー率（Quantum Bit Error Rate）
QKD	量子鍵配送（Quantum Key Distribution）
QKDN	量子鍵配送ネットワーク（QKD Network）
RNG	乱数生成器 (Random Number Generator)
XOR	排他的論理和 (Exclusive OR)

5. 表記法

本標準では、以下の表記法を用いる。

キーワード「が要求されている(is required to)」は、厳密に従わなければならず、この文書への適合性が主張される場合にはそこから逸脱することは許されない要求条件を示す。

キーワード「推奨される(is recommended to)」は、推奨されるが絶対に必要ではない要求条件を示す。したがって、この要求条件は、適合性を主張するために存在する必要はない。

6. 鍵管理の概要

本標準は、QKD 技術をサポートするネットワークの設計および実現に不可欠な QKDN の鍵管理について記述する。

QKD モジュールによって鍵が生成されると、暗号アプリケーションに供給されるまで、古典ビット列として QKD ノードにある KM と呼ばれるサーバに安全に格納される。KM は、許可されていない当事者による侵入および攻撃から保護されたトラステッドノードである QKD ノードに配置される。通常の場合、鍵のサイズは変更され、暗号アプリケーションに適切な長さにフォーマットされる。

ポイントツーポイント QKD リンクは、鍵供給の到達可能性と可用性を高めるため、トラステッドノードである QKD ノードを介して連結され、QKD ネットワーク(QKDN)を構成する。QKDN では、トラステッドノードを介して鍵をリレーし、QKD リンクで直接接続されていない場合でも、2 つの当事者間で共有することができる。鍵は、機能エンティティから別の機能エンティティへ転送されるときに、適切な方法で同期され、認証される。その後、鍵はユーザネットワークの様々な暗号アプリケーションに転送される。生成（量子レイヤからの受取）、格納、フォーマット、リレー、同期、認証、供給、削除/保存など鍵ライフサイクルにおいて実行されるすべての鍵に関する動作は、鍵管理と呼ばれ、QKD アプリケーションとサービスの中心に位置する。

QKDN の基本機能と階層構造は[ITU-T Y.3800]により定義される。図 1 は、QKDN における鍵管理の基本的な動作を示している。QKD リンクによって接続された QKD モジュールの対は、それぞれ独自の方法で鍵を生成する。生成された鍵は KM に転送される。KM は鍵を管理し、それらをユーザネットワークのサービスレイヤ内の暗号アプリケーションに提供する。鍵は、KM を介してリレーされ任意の QKD ノード間で共有することができる。QKDN コントローラは、鍵リレーのルーティング制御を実行する。QKDN マネージャは、QKD ネットワーク全体の状況を監視し、KM による鍵ライフサイクル管理と QKDN コントローラによる鍵リレーのルーティング制御と再ルーティング制御をサポートする。

QKDN モジュールは、量子ビットエラーレート(QBER)の増加などのアラームを検出すると、QKDN コントローラに対し直接または KM 経由で間接的にアラートを通知する。QKDN コントローラは、障害のある QKD リンクを迂回して鍵リレーのパスを再ルーティングし、最終的に鍵を供給する。

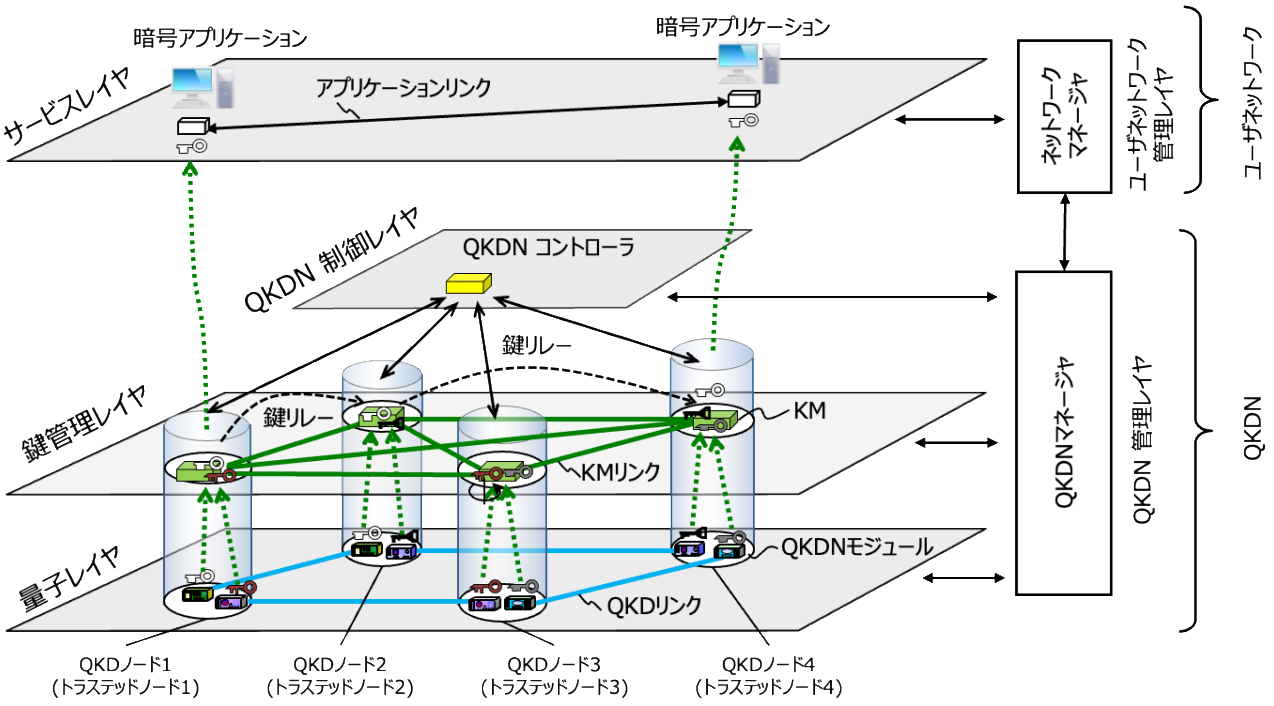

図 1　QKDN の鍵管理の基本動作

7. 鍵管理の機能要素

図２は、[ITU-T Y.3802]が規定する QKDN の機能アーキテクチャモデルを示し、鍵管理レイヤとそれに関連する参照点をハイライトしている。

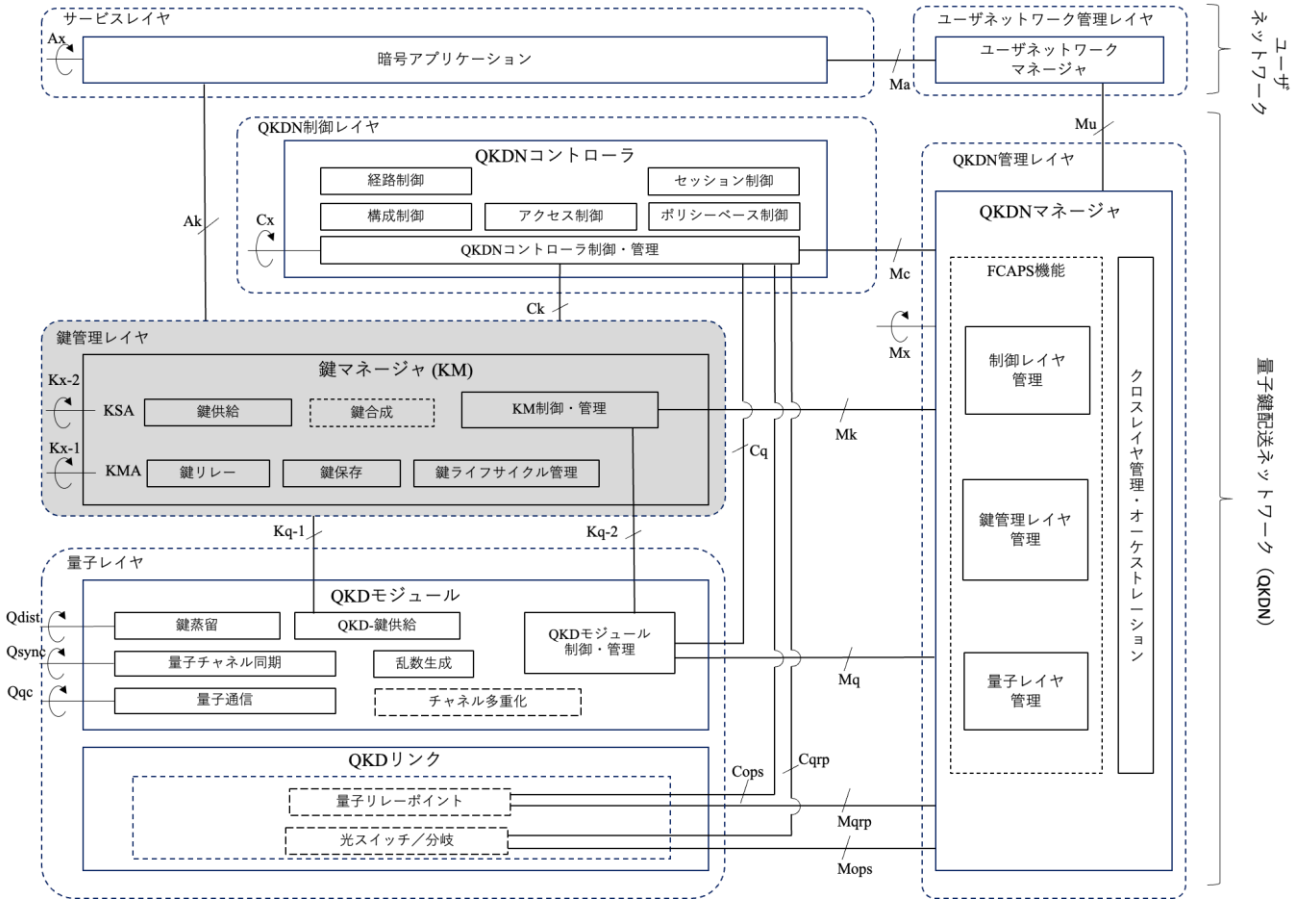

図２ QKDN のハイレベル機能アーキテクチャモデル

[ITU-T Y.3801]が規定する要求条件を満たす鍵管理機能を設計、実装および実行するためには、図３に示すように、KM 内の 2 つの能要素、すなわち鍵管理エージェント(KMA)と鍵供給エージェント(KSA)を識別して定義するのが簡便かつ実用的である。

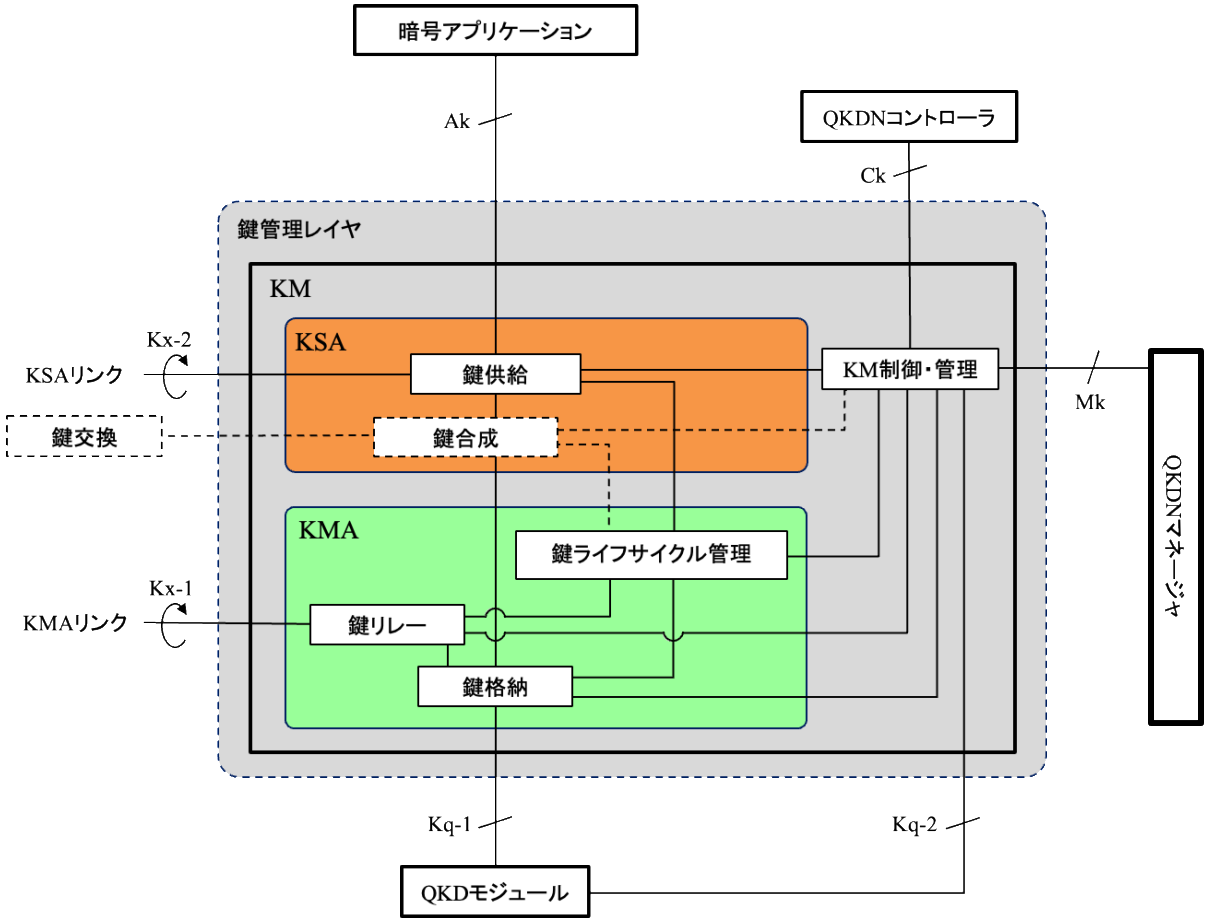

図３　鍵管理レイヤの機能アーキテクチャモデル

これらの機能要素の役割を以下に示す。

7.1.　KM内のエージェント

a)　KMA：QKD モジュールから鍵を取得し、QKD ノードを鍵リレーで相互接続し、[ITU-T Y.3801]の要求条件 Req_KM.1～9 および 11 で規定される以下の機能を実現する。

- 鍵格納
- 鍵リレー
- 鍵ライフサイクル管理

b)　KSA：KMA と暗号アプリケーションの間に位置し、暗号アプリケーションとインターフェースで接続し、[ITU-T Y.3801]の要求条件 Req_KM.10 で規定される以下の機能を実現する。

- 鍵供給

KSA は、さまざまな暗号アプリケーションをサポートするアプリケーションプログラムインターフェースのライブラリを収容する。オプションとして、[ITU-T X.1714]で規定される以下の機能を含む。

- 鍵合成

鍵合成機能は、KMA からの鍵と、鍵交換方式で提供される他の鍵とを、KMA からの入力鍵のセキュリティと同じセキュリティになるように合成して、鍵供給機能に出力する。鍵合成機能のセキュリティに関する詳細は、本標準の範囲外である。

c) KMA 及び KSA に加えて、KM は以下の機能を含む。

- KM 制御・管理

これは、[ITU-T3801]の要求条件 Req_KM4、5 および 9 を満たすために、QKD モジュール、QKDN コントローラおよび QKDN マネージャと通信を行う機能である。

7.2. KMリンク

２つの機能要素を識別したことで、KM リンクは KMA リンクと KSA リンクから構成される。KMA および KSA は、プロセッサ、ストレージ、および通信インターフェースを持つ。

注１- 実際の設定では、KMA と KSA、および KMA リンクと KSA リンクは、単一のサーバと単一のリンクで縮退的に実装することができる。

注２- 例えば、鍵リレーノードとしてのみ動作する場合など、QKD ノードが KMA のみを実装し、KSA を持たない場合がある。

7.3. 参照点

KM に関連する参照点(Ak, Kq-1, Kq-2, Kx-1, Kx-2, Ck, Mk)は[ITU-T Y.3802]で定義している。

7.4. セキュリティ分界点

セキュリティドメインは、情報セキュリティおよびネットワークセキュリティにおける一般的な概念である。[ITU-T Y.3800]では、供給される鍵に対する 1 つのレイヤの責任と、鍵の使用に対する別のレイヤの責任を区別する境界として定義される。単一のセキュリティポリシーの対象となるエンティティとパーティの集合を分割する。セキュリティドメインの境界は、セキュリティ責任分界点に相当することが多い。

分界点は、鍵の管理と利用の責任を分ける境界として KSA と暗号アプリケーションの間に置くことができる。この場合、暗号アプリケーションに鍵が提供されると、KMA と KSA は鍵管理ポリシーに従って鍵を削除または保存されるだろうが、一方で、暗号アプリケーションは自分自身の責任で鍵を使用する。暗号アプリケーションは 1 つの鍵を 2 回以上使用しないことが強く推奨される。

8. 鍵管理の詳細手順

図 4 は、リファレンス・モデルに基づいて鍵管理の手順を説明しており、どのように鍵を管理し、暗号アプリケーションへ供給するかを示す。

図 4 に示すケースでは、QKD ノード 1 と 3 は QKD リンクによって直接接続されていない。これらの 2 つのノード間で鍵を共有するために、QKD ノード 2 および 3 の QKD モジュールで生成された鍵を用いて、KMA2 と KMA3 の間で鍵リレーが実行される。最終的に KSA1 と KSA3 から、鍵を要求した暗号アプリケーションに対して鍵が供給される。QKDN コントローラは、鍵リレーのためのルーティング制御を実行する。QKDN マネージャは、QKDN 全体の状況を監視し、QKDN を管理し、必要に応じて KMA、KSA、QKDN コントローラをサポートする。

QKDN における相互接続性と拡張性を維持するために、様々なタイプの情報を含むメタデータが追加された鍵データの適切な鍵フォーマットを導入する必要がある。本標準では、鍵データとメタデータの論理的なセットを鍵ファイルと呼ぶ。

これに基づいて、KMA、KSA、暗号アプリケーションは、鍵データ、メタデータ、鍵管理情報を相互に通信する。さらに、QKDN コントローラおよび QKDN マネージャは、メタデータに基づいて、KMA および KSA と制御情報および管理情報を通信する。以下では鍵管理の基本的な手順について述べる。

セキュリティ上の問題と方法は、本標準の範囲外である。

注 1 - 図 4 では、水平の実線と垂直の実線矢印がそれぞれ KMA リンクとインターフェースパスを表し、これらのリンク上で鍵が伝達されるため、鍵の機密性、完全性、および真正性を考慮する必要がある。水平の点線は KSA リンクを表し、鍵同期と認証に関する情報が伝達される。暗号アプリケーションから KSA への垂直の点線矢印は、鍵の要求を表す。他の点線矢印は、接続されたエンティティ間で QKDN の制御情報および管理情報を通信するためのリンクを表す。これらの点線と点線矢印については、情報の完全性が最も重要である。

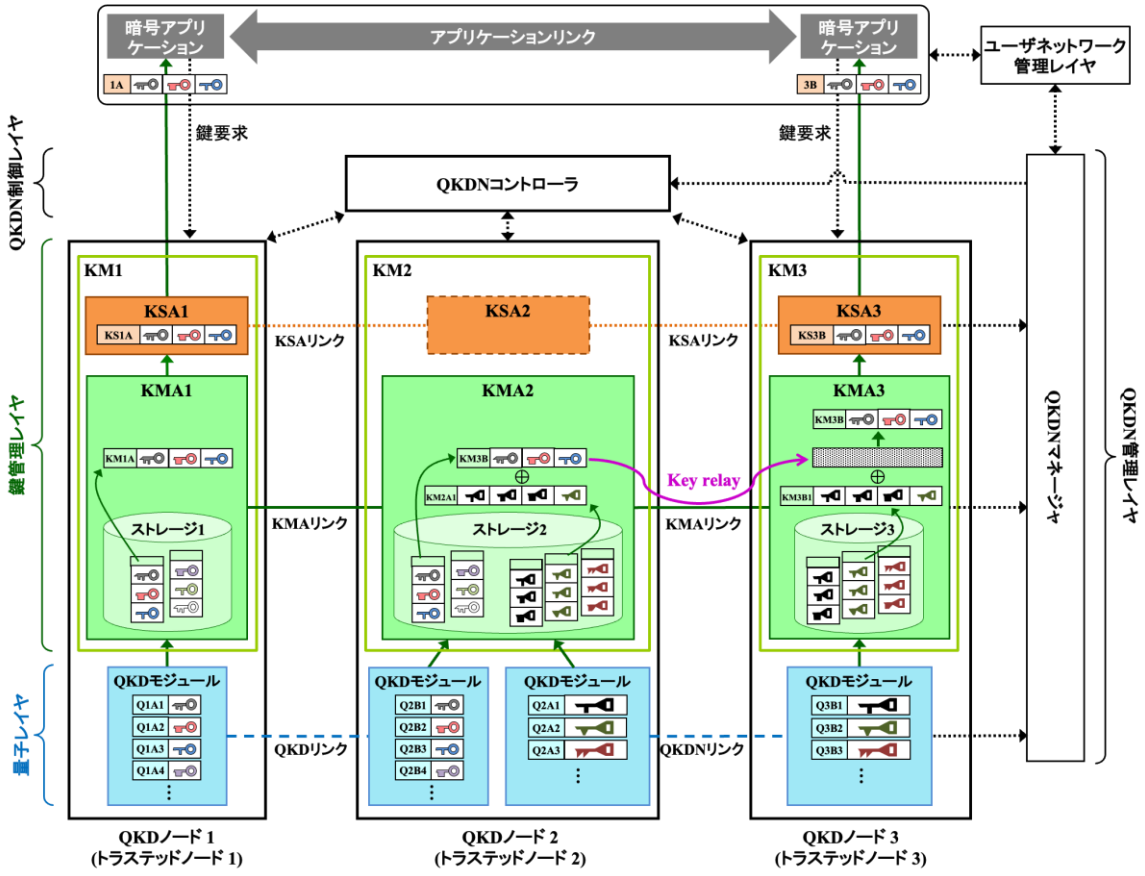

図 4　鍵管理の機能要素と詳細手順

注 2-図 4 は、3 つのノードを持つ最も簡単な QKDN を示している。このような簡略化においては、鍵リレーは表現できるものの、ルーティング制御の側面をうまく表現することができていないが、これを除外していない。

8.1.　量子レイヤにおけるQKD-鍵の生成

量子レイヤでは、一対の QKD モジュールが、QKD の IT セキュアなプロトコルに基づく独自の方法で一対の対称な(同一の)ランダムビット列を生成する。各 QKD モジュールは、セキュリティ脅威に対して安全で信頼性の高いノード(トラステッドノード)に設置される。

注 1 - 暗号の分野では、習慣的に、鍵または暗号文の送信者と受信者をそれぞれ「Alice」と「Bob」と呼ぶ。本標準では、鍵を共有する、QKD リンクによって接続された QKD モジュールを示すためにしばしば Alice と Bob を使用する。"prepare and measure"スキームと呼ばれる QKD プロトコルの場合、Alice と Bob はそれぞれ送信者と受信者となる。[ITU-T Y.3800]で言及されている MDI-QKD や TF-QKD のような測定支援方式に基づく QKD プロトコルの場合、Alice と Bob は両方とも送信者であり、受信者は量子チャネル上の中間点に位置する。量子もつれベースの QKD プロトコルの場合、Alice と Bob の両方が受信者であり、量子もつれ状態の量子信号の送信者が量子チャネル上の中間点に位置する。

注 2 - 本標準においては、QKD プロトコルとは、量子情報理論に基づく情報理論的セキュリティを有する対称暗号鍵を確立するための手順の一覧をいう。

QKD プロトコルは、QKD モジュールの対ごとに異なることがあり、各 QKD モジュールの対は異なるベンダによって独立に提供されることがある。以下では、QKD モジュールで生成された対称なランダムビット列を QKD-鍵と呼び、KMA でサイズ変更およびフォーマットされた鍵、および KSA 内にあるそれらの鍵とは区別する。異なる対となる QKD モジュールによって生成される QKD-鍵の単位長は、互いに異なる場合がある。

各 QKD モジュールではメタデータが生成され、QKD-鍵に付加されて鍵ファイルが形成される。QKD モジュールの対は、QKD-鍵ファイルを対応する KMA に転送する。

QKD-鍵のメタデータは、10 章表 1 の項目(1)に規定されている。

8.2. 鍵管理レイヤにおけるKMA-鍵とKSA-鍵の管理

この章では、KMA-鍵と KSA-鍵の 2 種類の鍵データを導入し、鍵管理レイヤで実行される動作手順を記述する。これには、6 つの主要な手順があり、以下の節で説明する。

これらの節の順序は、実際の実行の時間的順序と必ずしも一致せず、実際の状況やユースケースによって異なる場合がある。

8.2.1. KMAにおける鍵取得、認証、格納

[ITU-T Y.3801]の Req_KM2 で要求されているように、KMA は、同じ QKD ノードにある QKD モジュールから、参照点 Kq-1 のインターフェースを介して QKD-鍵ファイルを受信し、格納が必要なときはそれらを安全に格納する。取得した QKD-鍵ファイルの長さは、互いに異なる場合がある。従って、[ITU-T Y.3801]の Req_KM3 で推奨されているように、KMA は、QKD-鍵を所定の単位長の鍵に再フォーマット(結合または分割)し、一時的にバッファに格納する。

バッファされた鍵を鍵データとして格納する前に、一対の QKD-鍵を受信した一対の KMA(KMA1 および KMA2 とする)は、バッファされた鍵の同一性を確認する。したがって、[ITU-T Y.3801]の Req_KM8 で推奨されているように、KMA は鍵同期、エンティティ認証、およびメッセージ認証の能力を持つ。KMA の対は、KMA リンクを介して相互に認証する。次に、KMA の 1 つ(KMA1)は、組となる KMA（KMA2）に、バッファされた鍵のハッシュ値またはメッセージ認証コードを含む鍵認証要求を送信する。次に KMA2 は、バッファされた鍵を(ビット位置で)同期し、手元にあるハッシュ値またはメッセージ認証コードを KMA1 から送られてきたものと比較することによって認証する。ハッシュ値の通信に関するセキュリティの詳細は、本標準の範囲外である。ハッシュ値またはメッセージ認証コードが一致する場合、KMA1 と KMA2 は最終的に、バッファされた鍵をメタデータと共に鍵格納ディレクトリに鍵データとして格納し、この鍵を KMA-鍵と呼ぶ。一致しない場合は、KMA1 と KMA2 はバッファされた鍵を廃棄する。

KMA-鍵のメタデータは、10 章表 1 の項目(2)に規定されている。

8.2.2. 暗号アプリケーションからの鍵要求の受信

サービスレイヤの暗号アプリケーションは、KSA に鍵要求を送信する。暗号アプリケーションからの鍵要求は、鍵供給サービスポリシーなどに応じて、必要なセキュリティレベルに関する情報を含むことができる。[ITU-T Y.3801]の Req_KM10 で要求されているように、KSA は、許可された暗号アプリケーションから、参照点 Ak の鍵供給インターフェースを介して鍵要求を受信する。KSA による鍵要求の受信の制御は、特に、鍵要求が複数の暗号アプリケーションから送信される場合、QKDN コントローラによってサポートされる。

次に、KSA は、適切な手段によって暗号アプリケーションを認証する。それらの証明書は、暗号アプリケーションおよび KSA を含む登録された機能要素のアクセス制御リポジトリを管理する QKDN コントローラのアクセス制御機能によって発行することができる。KSA と暗号アプリケーションは、認証を確立すると、次回の鍵要求の認証で用いる共通の秘密鍵を交換し共有できる。

注 - KSA と暗号アプリケーションは、鍵の一部を次の認証の秘密鍵として保持できる。

KSA が暗号アプリケーションを認証した後、KSA と同じノードの KMA は互いを認証することが推奨される。認証後、KSA は、KMA に、個々の鍵要求で要求された情報（例えば、鍵長、鍵量、ノード対名または ID、および KSA-鍵 ID）を通知する。KSA-鍵 ID は、QKDN 全体で一意の ID である。

8.2.3. KMA間の鍵リレー

[ITU-T Y.3801]の Req_KM6 で推奨されているように、KMA は、高度に安全な暗号化(例えば、OTP[b-Shannon1949])を使用して、2 つのエンドポイント KMA 間の鍵リレールートを経由する鍵リレーをサポートする。鍵リレールートは、図 1 に示すように、QKDN コントローラによって制御される。エンドツーエンド鍵を確立するための KMA 間の通信セッションは、QKDN コントローラによって制御することができる。

鍵の機密性を確保するために IT セキュアプロトコルである OTP を用いてのポイントツーポイント鍵リレーの典型例を以下に説明する。

KMA-鍵の鍵データとメタデータを、隣接する QKD モジュールの対が共有する他の鍵と OTP 方式で排他的論理和(XOR)を行い、送信元 KMA から送信先 KMA に送信することで、IT セキュアな鍵リレーを実現する(図 4 参照)。送信先 KMA で復号した後、送信元 KMA、送信先 KMA 及び鍵リレータイムスタンプからなる鍵リレー情報を送信元の KMA-鍵メタデータに付加して、リレーされた KMA-鍵のメタデータとして送信先 KMA の鍵ストレージに格納する。

鍵をさらに次の送信先ノードにリレーする場合、鍵リレー情報を含めて前回と同様に暗号化する。この場合、鍵リレー情報は、現在のノードを送信元 KMA とし、次の送信先ノードを送信先 KMA とし、現在のノードにおける新しいリレータイムスタンプで更新される。この更新された鍵リレー情報は、送信元の KMA-鍵メタデータに付加され、第 2 の送信先 KMA に格納される。

鍵リレーには、KMA に備えられた特別な乱数生成器(RNG)を使用するという別の方法もある。送信元 KMA 内の RNG は、乱数列を生成する。送信元 KMA は、その乱数列をストレージに格納し、そのコピーを OTP ベースの鍵リレーによって送信先 KMA にリレーする。これにより、対称なランダムビット列の対をエンドポイント KMA 間で共有することができる。

注 1-図 4 では、KMA1 と KMA2 はそれぞれ「KM1A」と「KM3B」というメタデータを持つ鍵ファイルとして、ストレージから必要な量の鍵を読み込む。KMA2 では、後者の鍵ファイルは、 メタデータ「KM2A1」を持つ他の鍵(隣接する QKD モジュールの対によって共有される)と OTP 方式で XOR によって暗号化され、KMA リンクを介して KMA3 に送信され、最後にメタデータ「KM3B1」を有する鍵で復号される。以上で鍵が KMA1 と KMA3 の間で共有される。

注 2-図 4 に示すスキームでは、鍵リレーは鍵ファイル(メタデータと鍵データ)の単位で実行される。なぜなら、鍵リレーの実装が単純化でき、メタデータのサイズは長くなく、通常は数 10 バイトであるためである。例えば、リレーされる鍵ファイル全体(メタデータ「KM3B」と鍵データ)は、他の鍵データ(メタデータ「KM2A1」を持つ鍵データ)を用いた OTP で暗号化される。しかし、メタデータを KMA-鍵によって OTP 方式で暗号化しないオプションは除外しない。

注 3 - RNG は、非決定論的であるべきである。これは、[b-ISO/IEC18031]に規定されているような従来の物理的雑音に基づく方式、又は量子原理に基づく方式(量子雑音 RNG)で実現することができる。

[ITU-T Y.3801]の Req_KM7 で推奨されているように、KMA は、OTP 暗号化方式に加えて、鍵管理ポリシーに従って鍵リレーのための別の暗号化方式(例えば、AES[b ISO/IEC18033-3]、[b-FIPS PUB 197])をサポートする。例えば、OTP 暗号化鍵リレーに必要な鍵量がない場合には、AES などの対称鍵暗号によるバックアップ方式を鍵リレーに用いることができる。

Req_KMA1　KMA では、個々の鍵リレーで使用される暗号化方式を記録するためのメタデータを作成し、このメタデータをリレーされた KMA-鍵の鍵ライフサイクル管理に使用することを推奨する。

各鍵リレーにおいて、送信元と送信先 KMA が共有する鍵の内容(同一性)が改変されていないか確認することが求められる。したがって、[ITU-T Y.3801]の Req_KM8 で推奨されているように、KMA の真正性及び KMA-鍵の完全性を保証するために、適切なエンティティ認証とメッセージ認証を使用する。特に、KMA-鍵の完全性保護のために、KMA-鍵データのハッシュ値やメッセージ認証コードを使用することができる。

KMA 間で交換される鍵管理情報の完全性は、メッセージ認証を実行することによって保護される。真正性と完全性を確保するための詳細なオプションは、本標準の範囲外である。

リレーされる KMA-鍵のメタデータは、10 章表 1 の項目(3)に規定される。

8.2.4.　KSAから暗号アプリケーションへの鍵供給

KSA と KMA が相互に認証した後、KSA は、要求された情報 （例えば、鍵長、鍵量、ノードペア名または ID、KSA-鍵 ID、および鍵のセキュリティレベル）を KMA に通知する。KMA は、KMA-鍵データのストレージから必要な量の鍵を取得する。このとき、オプションとして、要求されるセキュリティレベルと鍵供給ポリシーに基づき、鍵リレー暗号化方式のメタデータを考慮する。

Req_KMA2　暗号アプリケーションが QKD-鍵と同じセキュリティレベルの鍵を要求する場合、KMA は、鍵リレー暗号化方式のメタデータを用いて、OTP 暗号化方式によってリレーされた KMA-鍵を選択することが推奨される。

次に、KMA は、この鍵を KSA に転送する。KSA は、図 4 の QKD ノード 1 および 3 に示されているように、この鍵を受信する。

注 - KSA から要求された長さに従って KMA 内の鍵データサイズを調整するために、時には KMA-鍵データが分割され、一部が KSA に供給される場合がある。この場合、残りの鍵データは、KSA からの次の鍵要求に使用されることがある。そのため、元の KMA-鍵 ID に加えて、残りの鍵データを識別するための別の情報が必要である。この目的のため、残りの鍵データから一部のビット列を取り出し、新しい KMA-鍵 ID として用いることができる。

KSA が取得した鍵を KSA-鍵と呼ぶ。[ITU-T-X.1714]で記述されているように、KSA では、オプションとして、取得した KMA-鍵データを、鍵交換方式によって提供される別の鍵と合成し、出力する KSA-鍵として生成することができる。

鍵が KMA から KSA へ転送された後、KMA はそのメタデータをストレージに記録し、[ITU-T Y.3801]の Req_KM11 と Req_M9 で要求されているように、鍵ライフサイクル管理のためにそのメタデータを QKDN マネージャに送信するかもしれない。KSA では、鍵を供給した後、暗号アプリケーション名、アプリケーションの送信元と送信先 ID、KSA-鍵 ID、KSA-鍵長、供給タイムスタンプなどの KSA メタデータを保持することができる。

最後に、[ITU-T Y.3801]の Req_KM８で推奨されているように、ハッシュ値またはメッセージ認証コードが、個々の鍵要求に応じてノード対の各 KSA 内の KSA-鍵データから計算される。KSA の対は、KSA リンクを介してそれぞれのハッシュ値またはメッセージ認証コードを KSA-鍵 ID と併せて比較し、KSA-鍵を同期して認証する。ハッシュ値通信のセキュリティに関する詳細は、本標準の範囲外である。

上記の検証が完了すると、[ITU-T Y.3801]の Req_KM10 で要求されているように、KSA-鍵 ID を持つ鍵データが、参照点 Ak の鍵供給インターフェースを介して暗号アプリケーションに供給される。KSA メタデータは、[ITU-T Y.3801]の Req_KM11 と Req_M9 で要求されているように、KSA のストレージへ記録、および／または鍵ライフサイクル管理のために QKDN マネージャへ送信される。暗号アプリケーションは、KSA-鍵 ID に基づいて鍵データを識別し、それを使用する。

暗号アプリケーションに鍵が提供されると、KMA と KSA は、[ITU-T Y.3801]の Req_KM10 で要求されているように、例えば鍵データをストレージから削除したり、鍵データをストレージに保存したりするなど、鍵管理ポリシーを適用しなければならない。

KSA-鍵のメタデータは、10 章表 1 の項目(4)で規定される。

鍵供給のためのセッションの制御は、特に鍵が複数の暗号アプリケーションに供給されるとき、QKDN コントローラによってサポートされることができる。

8.2.5. KMA間の鍵リレーの再ルーティング

鍵リレーの再ルーティングは、鍵供給/鍵生成の継続的な可用性を保証するために、鍵管理レイヤおよび/または量子レイヤの状態に応じて実行されなければならない。典型的なケースとしては、リレーノード内の KMA-鍵の量が閾値を下回る場合、および高い QBER により特定の QKD リンクによって接続されたリレーノード間での QKD-鍵生成の妨げられる、またはその生成レートが低減する場合が挙げられる。詳細は[ITU-T Y.3804]を参照のこと。

[ITU-T Y.3801]の Req_KM5 で要求されているように、KMA は、鍵管理に関する情報を KM 制御・管理機能経由で QKDN コントローラと QKDN マネージャに提供する。

[ITU-T Y.3801]の Req_KM4 で推奨されているように、KM 制御・管理機能は、量子レイヤ内の QKD モジュールから QKD モジュールのステータス情報を、オプションとして QBER、鍵レート、QKD リンクステータス、障害発生時のアラームなどの QKD リンクのステータス情報を受信する。

QKDN コントローラは、KMA から鍵消費率、KMA 内の鍵の共有量、KMA リンクステータスなどの鍵管理に関する情報を取得し、QKD モジュールから QKD モジュールのステータス情報およびオプションとして QKD リンクのステータス情報を取得する。

特に、リレーノードに接続された QKD リンクについてのアラームが KMA に通知されると、KMA はそのアラームを QKDN コントローラに転送し、KMA は QKDN コントローラが提供する再ルーティングに従って鍵リレーを継続する。

QKDN コントローラから指示される再ルーティングの方法には、次の 4 つが挙げられる。

a) 手動型：鍵リレールートが KMA に手動で設定される。

b) 固定レート型：QKD リンクの鍵生成レートに基づいて、ノード対の鍵リレーの必要頻度を見積り、鍵リレー指示リストを作成する。KMA はこのリストに従って鍵リレーを行う。

c) データトラフィック適応型：KMA は、鍵リレートラフィックの直近の統計記録に基づき自動的に鍵リレーを実行する。

d) スケジュール型：ある条件(例えば、鍵消費および/または鍵生成率等)の変化を予測できる場合に、鍵リレーを予め計画する。

8.2.6. 鍵ライフサイクル管理

[ITU-T Y.3801]の Req_KM11 で要求されているように、各 KMA は、鍵の受信、格納、フォーマット、リレー、同期、認証、供給、削除/保存など、KMA が行う鍵管理動作に関する情報を格納する。例えば、各 KMA はメタデータを収集し、KMA-鍵ファイルに格納し、QKDN のマネージャに送信する。各 KSA は、メタデータを KSA-鍵ファイルに格納し、QKDN マネージャに送信する。

KMA は、鍵ライフサイクルにおけるすべてのフェーズ移行を、これらのワークフローを監視、監査、追跡する QKDN マネージャと連携して処理する。

QKDN マネージャの障害管理機能は、KMA-鍵が許可されていない当事者に不正に漏洩したり、アクセスされた疑いがあるなど想定外の障害が検出された場合には、鍵ライフサイクル管理情報を追跡して原因を分析し、QKDN コントローラおよび KMA による対策の実行を指示する。最悪の場合、KMA は、セキュリティ侵害の疑いのある鍵と相互関係を持つ可能性のある関連する鍵を削除する。

9. 鍵リレーのいくつかのスキーム

送信元ノードと送信先ノードとの間で鍵を共有する鍵リレースキームを図５に示す。Key_{AB} は、KMA-A と KMA-B の間で生成される。Key_{AB} は、Key_{BC} を用いて OTP 暗号化され KMA-B から KMA-C にリレーされる。Key_{AB} は、Key_{CD} を用いて OTP 暗号化され、KMA-C から KMA-D にリレーされ、最終的に KSA-D に供給される。

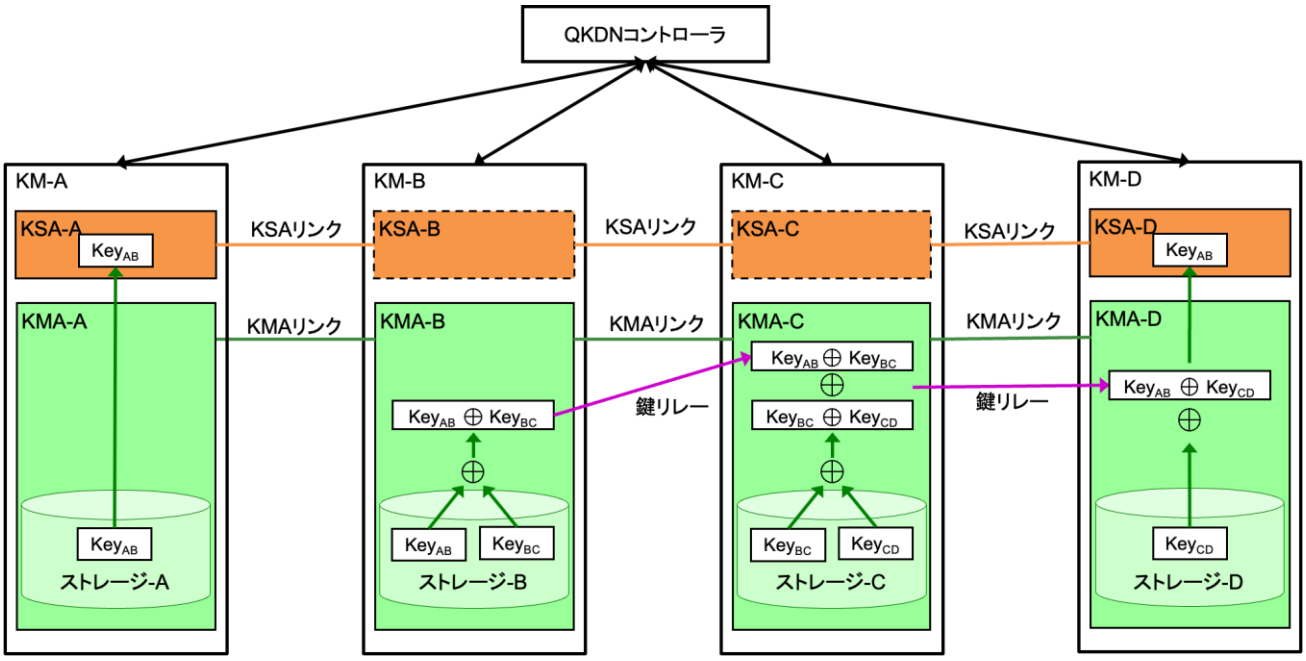

図５　[ITU-T Y.3800](ケース 1)におけるポイントツーポイントの鍵リレースキーム

図６に示されるケース２では、KMA-A から KMA-D への鍵リレーに、KMA-A でローカルに生成されたランダムビット列 Key_{RN} が使用される。

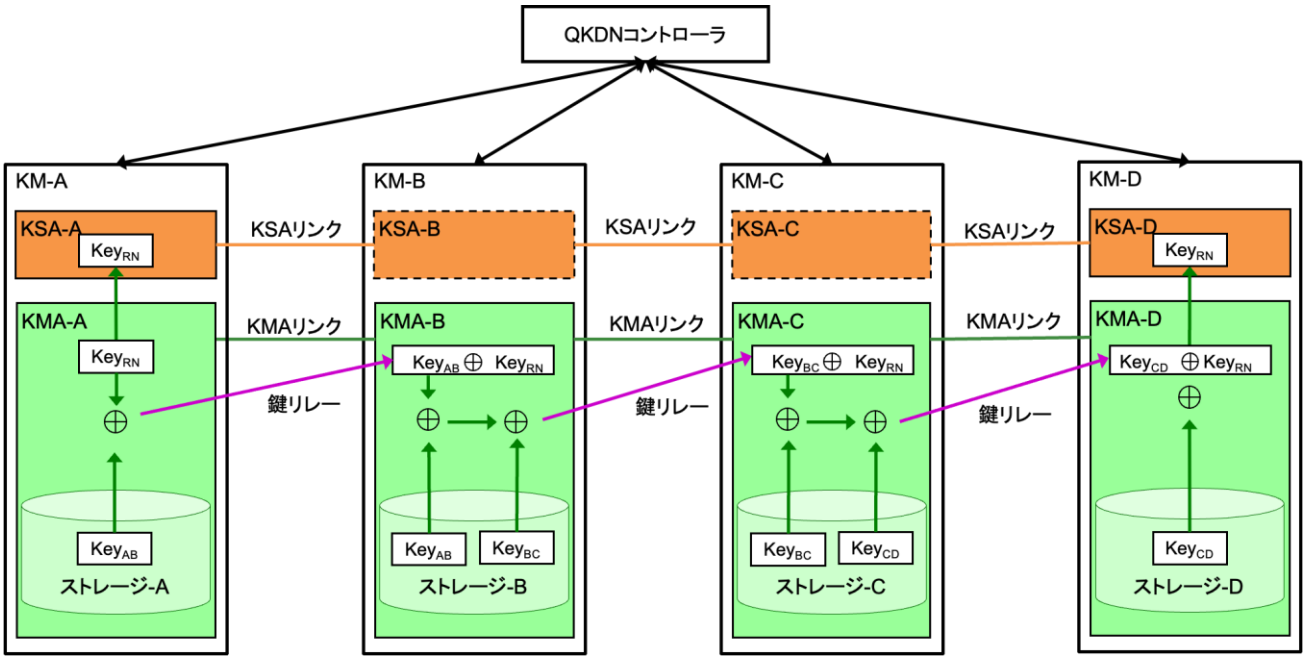

図６　[ITU-T Y.3800(ケース 2)]のポイントツーポイントの鍵リレースキーム

実際のアプリケーションにおいては、ポイントツーポイント鍵リレースキームのケース１およびケース２には２つの問題がある。１つは、XOR 暗号鍵とそれに対応する復号鍵が１つのリレーノード上に共存し、それはセキュリティリスクを有する。もう１つは、ネットワークがノード間の複雑な KMA リンクを持ち、結果として実装が困難になることである。上記の問題を緩和するため、修正した鍵リレースキームを２つ、以下で記述する。

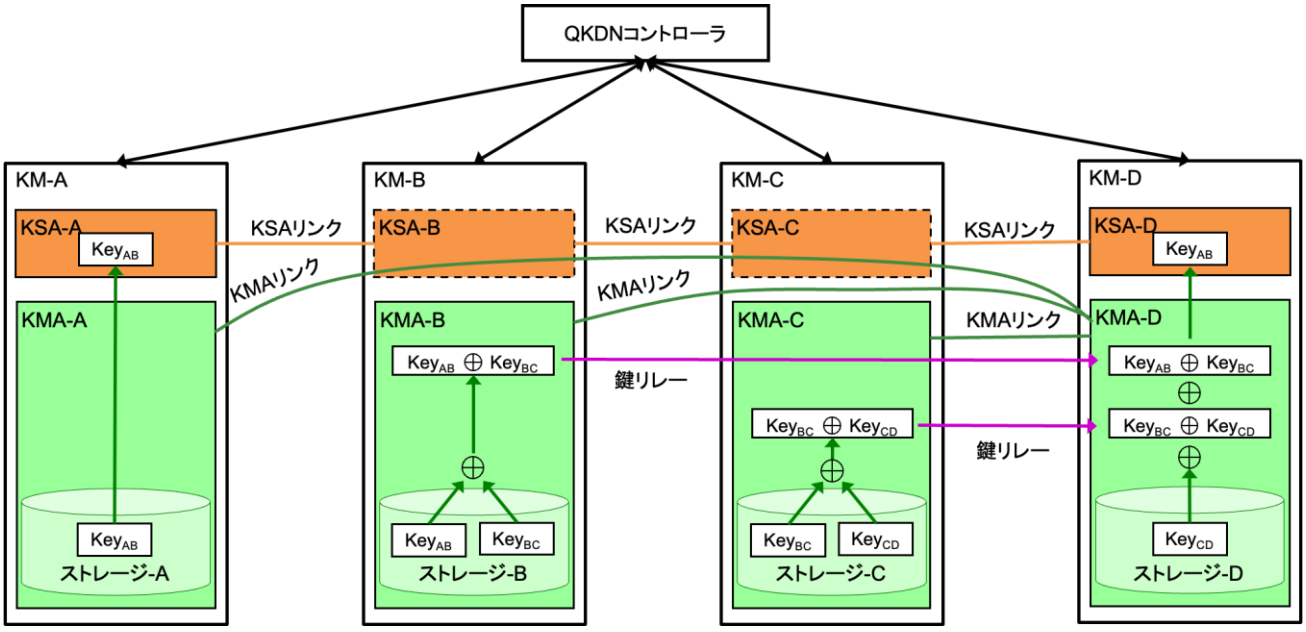

図７　送信先ノードで集中的に XOR 暗号文を処理する鍵リレースキーム

図５のスキームと対比し、図７のスキームは次のように修正される。

1)　ノードの KMA リンクは、隣接ノードではなく、送信先ノードに接続する。

2)　XOR 暗号鍵は、送信先ノードに直接送信する。

3)　送信先ノードにおいて、受信したすべての暗号文を XOR することによりリレーされる鍵を復号する。

この方式では鍵リレーノードの機能が簡素化される。XOR 暗号鍵のみが鍵リレーノードによって送信され、隣接ノードでのリレーは必要とされない。このようにすると、リレーされる鍵がルートに沿ってノードを通過することがなく、その XOR 暗号文が対応する復号鍵と同一ノード上で共存することがない。

ただし、このスキームではネットワーク内の複雑な KMA リンクが単純化されない。そのため、リンク長を拡張するための幹線ネットワークなどの特定のシナリオにのみ適用される。

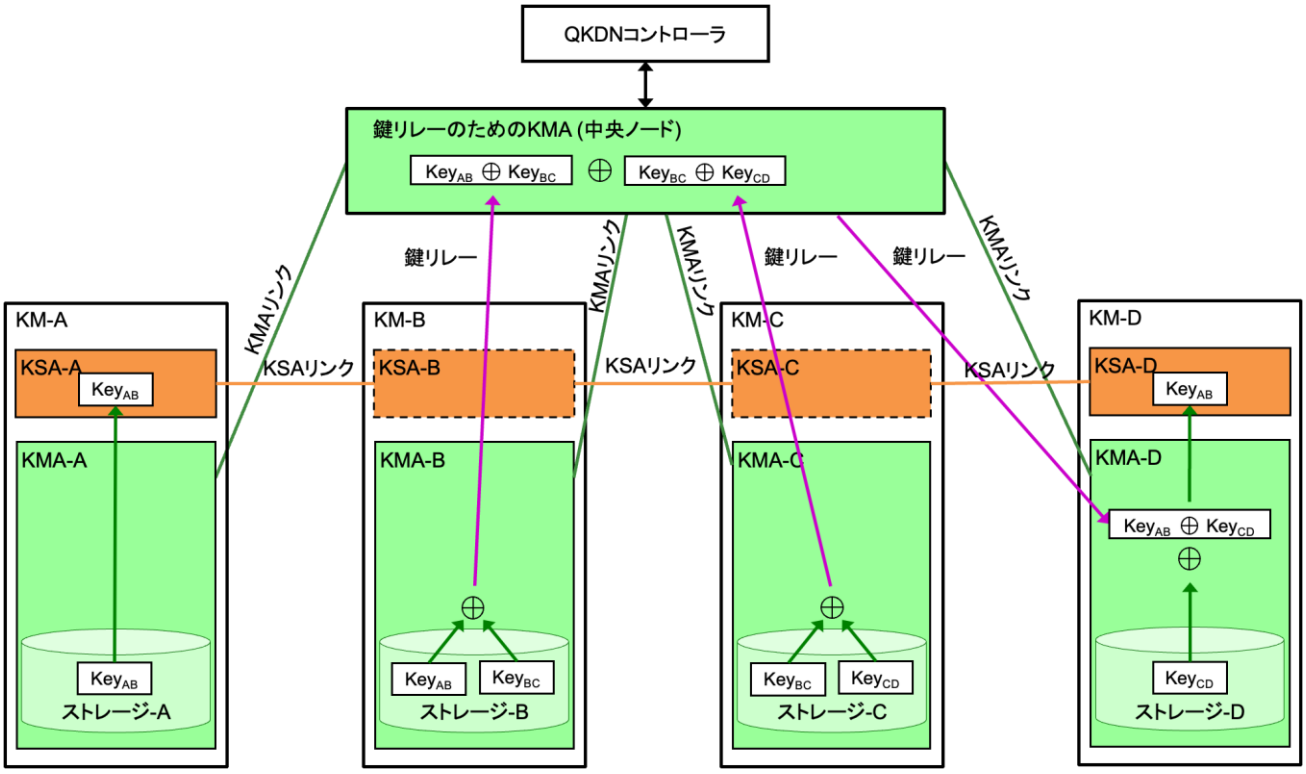

図８　中央ノードが XOR 暗号文を収集する鍵リレーのスキーム

図 7 のスキームとの対比で、図８のスキームは次のように修正される。

1)　すべての XOR 処理を中央ノードで行う。XOR 暗号鍵は、中央ノードが送信先ノードへ送信する。

2)　KMA リンクは、各 KMA と中央ノードの KMA の間に設置する。

3)　QKDN コントローラリンクは、QKD コントローラと中央ノードの KMA の間に設置する。

このスキームにより KMA リンクが単純化され、KMA リンクはノードと中央ノードとの間にのみ存在し、ネットワーク実装が容易となる。

10.　鍵ファイルフォーマット

鍵ファイルは、所定の大きさの鍵データと、鍵管理に必要な項目を含むメタデータから構成される。メタデータの内容は、QKDN のアーキテクチャ、例えば集中型アーキテクチャであるか分散型アーキテクチャであるか、およびユースケ

ースによって異なる。鍵ファイルのメタデータは、例えば各 KMA や KSA に分散的に格納することもできる。表 1 は、メタデータの基本的な項目をまとめている。

表１　メタデータ情報(基本情報)

メタデータ	記述	M/O
(1) QKD-鍵		
QKD-鍵 ID	QKD-鍵の識別子	M
鍵長	QKD-鍵の長さ	O
QKD モジュール ID	QKD-鍵を生成した QKD モジュール (Alice または Bob) の識別子	O
組合せ QKD モジュール ID	Alice と Bob の対を構成する組み合わされた QKD モジュールの識別子	O
生成タイムスタンプ	QKD モジュール対で QKD-鍵が生成された時刻のタイムスタンプ	O
ハッシュ値	QKD-鍵データのハッシュ値 (ハッシュ関数にはいくつかのオプションがあり、他の標準で検討される)	O
(2) KMA-鍵		
KMA-鍵 ID	KMA-鍵の識別子。Alice と Bob 向けの対となる鍵の識別子で、1 つの QKDN 内で一意である。QKD モジュールの対の名前から生成されたハッシュ値の一部が、しばしばこの識別子に使用される。	M
鍵長	KMA-鍵の長さ	O
鍵タイプ	暗号鍵か復号鍵かを指定する指標	O
KMA ID	KMA-鍵を格納する KMA の識別子	O
組合せ KMA ID	組み合わされる KMA の識別子	M
生成タイムスタンプ	KMA で KMA-鍵が生成された時刻のタイムスタンプ	O
QKD モジュール ID	KMA-鍵データに対応する QKD-鍵を生成した QKD モジュールの識別子	O
組合せ QKD モジュール ID	Alice と Bob の対を構成する組み合わされる QKD モジュールの識別子	O
ハッシュ値	KMA-鍵データのハッシュ値（ハッシュ関数にはいくつかのオプションがあり、他の標準で議論される）	O
(3)リレーされた KMA-鍵		
送信元 KMA ID	鍵リレーの送信元 KMA の識別子	O
送信先 KMA ID	鍵リレーの送信先 KMA の識別子	O
鍵リレータイムスタンプ 鍵リレー暗号化方式	鍵リレーのタイムスタンプ 鍵リレーに用いられた暗号化方式	O O
KMA-鍵メタデータ	送信元 KMA の KMA-鍵のメタデータ	M
(4) KSA-鍵		
KSA-鍵 ID	KSA-鍵の識別子	M
鍵長	KSA-鍵の長さ	O
供給タイムスタンプ	KSA から暗号アプリケーションへ KSA-鍵が供給された時刻のタイムスタンプ	O
アプリケーション名	暗号アプリケーションの名前	O
アプリケーション送信元 ID	暗号アプリケーションの送信元 ID	O
アプリケーション送信先 ID	暗号アプリケーションの送信先 ID	O

O:オプション項目、M:必須項目

注-今後、ユースケースの精査を経て、メタデータ構造の詳細が記述される予定である。また、キューの優先制御、サービス品質制御などの新たなメタデータ項目が拡張機能として記述される予定である。

参考文献

[b-ETSI GR QKD 007] ETSI Group Specification GR QKD 007 (2018), *Vocabulary*

[b-ETSI GS QKD 008] ETSI Group Specification GS QKD 008 (2010), *QKD module security specification*

[b-ISO/IEC 18033-3] ISO/IEC 10833-3:2010 (2010), *Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers.*

[b-FIPS PUB 197] Federal Information Processing Standards Publication 197 (2001), Announcing the ADVANCED ENCRYPTION STANDARD (AES)

[b-Shannon 1949] Claude Shannon, *Communication Theory of Secrecy Systems,* Bell System Technical Journal, vol. 28, pp. 666–682, 1949.