Vol.2021-CSEC-95 No.19
Vol.2021-SPT-45 No.19
Vol.2021-EIP-94 No.19
2021/11/9

セキュア光トランスポートNWの実現に向けたアーキテクチャ設計および形式検証による安全性評価

前田 志温[1]　中林 美郷[2]　奥田 哲矢[2]

概要：本研究では，今後のデータセンタ間通信の主流と期待されるセキュア光トランスポート NW（以降 SOTN）について，そのアーキテクチャの設計および形式化と，想定される攻撃者モデルに対する対策の提案，形式検証による安全性評価を行った．アーキテクチャ設計・形式化においては，従来の拠点間通信の事実上標準である IPsec を参考として，安全性評価においては，暗号利用プロトコルの形式検証ツールの事実上標準である ProVerif を利用した．本研究の結果，想定される攻撃者モデルにおいて，既存の鍵交換装置や認証暗号装置の継ぎ接ぎのみでは，未知の攻撃（Key Compromized Disagreement (KCD) Attack と呼ぶ）が検知されたこと，および，提案する対策により，左記の攻撃の影響を低減できることを示した．上記の結果は，個別に安全性が保証された装置であっても，それらの組合せ設計およびアーキテクチャ設計により，未知の攻撃が発生し得ることを示唆しており，さらに，Passive Adversary を許さない仕組み（具体的には量子鍵交換）を一部の通信路に適用することで，アーキテクチャ全体の安全性が向上するという，従来の情報セキュリティの常識で捉えられない対策が可能であることを示唆するものである．

Architecture Design and Security Evaluation with Formal Verification for Secure Optical Transport Network

1. 背景

近年，データセンタ間通信に要求されるスループットおよびレイテンシの水準が高まっており，大容量および低遅延な光トランスポート NW の導入が今後ますます進行すると予想されている [29, 30]．光トランスポート NW は，光ファイバ内で信号の波長分割多重化を行うことで大容量のスループットを，信号処理装置の光電変換を低減することで低遅延のレイテンシを実現できる．光トランスポート NW のセキュリティ対策としては，レイヤ１（物理層）で認証暗号を利用する OTNsec とレイヤ２（データリンク層）で認証暗号を利用する MACsec [11] が存在する．MACsec では，さらに鍵交換の仕様である MACsec Key Agreement が存在する [12].

一方で，近年は量子コンピュータの研究開発の進展により，従来の情報セキュリティの要である公開鍵暗号/鍵交換アルゴリズムについて，その安全性を帰着させる素因数分解問題や離散対数問題の効率的な多項式時間解法アルゴリズムの実現が危惧されている．そのため量子コンピュータに対しても安全な鍵交換の仕組みとして，量子鍵交換（Quantum Key Distribution, QKD）や耐量子計算機暗号（Post-Quantum Cryptography, PQC）の技術開発が進んでいる．光トランスポート NW のような耐用年数の長い基盤・インフラ的な装置・設備について，量子コンピュータによる暗号危殆化を見据えた仕組みへ移行することは自然である．

以上より，本論文では，量子コンピュータに対して安全な光トランスポート NW を実現するために，OTNsec, MACsec のような認証暗号を使った伝送の仕組みに，QKD, PQC のような鍵交換の仕組みを安全に適用することを目指す．本稿の構成を以下に示す．まず２章でアーキテクチャ設計のベースとなる前提知識について概説し，３章で IPsec を SOTN に適用する際の課題を述べる．４章で課題への対策として，提案プロトコルとその安全性要件の詳細を述べ，５章でその安全性評価について述べる．最後に６章で関連研究を，７章でまとめと今後の課題を述べる．

[1] 東京大学
Tokyo Uniersity
[2] NTT 社会情報研究所
NTT Social Informatics Laboratories

ⓒ 2021 Information Processing Society of Japan

Vol.2021-CSEC-95 No.19
Vol.2021-SPT-45 No.19
Vol.2021-EIP-94 No.19
2021/11/9

2. 前提知識

2.1 IPsec

IPsec はデータセンタ間通信を含む拠点間通信の事実上標準である．本節では，次章以降の SOTN のアーキテクチャ設計に先立って，設計のベースとする IPsec の仕組みを概説する．IPsec は，インターネットの事実上標準を定める IETF (Internet Engineering Task Force) で仕様化されている拠点間のセキュアチャネルを構成するプロトコルである．仕様書の構成としては，仕様の全体像を記述したアーキテクチャ [16]，暗号化 [15]，認証 [14]，鍵交換 [13] で構成される．

アーキテクチャは，鍵交換を行う IKE (Internet Key Exchange) と暗号化と認証を行う ESP (Encapsulated Security Payload)，認証を行う AH (Authentication Header) で構成される．「認証」には，メッセージ認証とエンティティ認証が含意される．それぞれ，通信先から届いたメッセージが正しいか検証すること，通信先が正しいか検証することに相当する．さらに，アーキテクチャの記述には，IKE で合意された鍵を ESP または AH に伝える SAD (Security Association Database) が含まれる（図 1）．

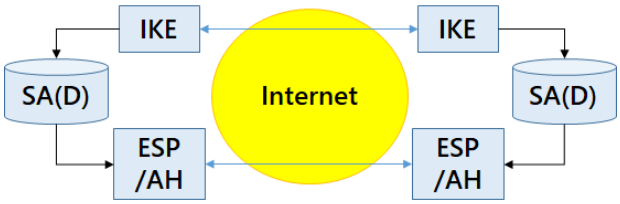


図 1: IPsec のアーキテクチャ．

SAD には，IKE/鍵交換時にネゴシエーションされた ESP/暗号化に必要な属性情報 (Security Associtation, SA) が入っている．下記に SA の具体例を示す．

- SA : Security Association 具体例
  - ID に相当する情報
    - SPI (Security Parameter Index)
    - Src/Dst IP Address (Port, Protocol)
  - 鍵に相当する情報
    - Key
    - Mode (ESP/AH, Tunnel/Transport)
    - Algorithm (IKE, ESP, AH, それぞれで使用する暗号アルゴリズム)
    - Other (Lifetime, MTU, etc.)

Protocol (IKE)．A, B 間での IKEv2 による鍵交換プロトコルを以下に示す．はじめに，鍵材料 X, Y とナンス N_A, N_B および属性情報を交換する．この鍵材料とナンスから A と B は共通鍵 K_{shared} を生成する．その後 A と B の署名鍵 Sig_A^{KD} と Sig_B^{KD} を用いて署名された認証情報を交換し，公開されている検証鍵 Ver_A^{KD} と Ver_B^{KD} を用いて相互認証を行う．prf(·) は疑似ランダム関数を表す．なお，本プロトコルにおいてはヘッダーやトラフィックセレクタなどいくつかのペイロードを省略している．

1. $A \to B$: SA_{A1}, X, N_A
2. $B \to A$: SA_{B1}, Y, N_B
3. $A \to B$: $\mathrm{SK}\{ID_A, ID_B, AUTH_A, SA_{A2}\}_{K_{shared}}$
4. $B \to A$: $\mathrm{SK}\{ID_B, AUTH_B, SA_{B2}\}_{K_{shared}}$

$K_{shared} \leftarrow \mathrm{keygen}(X, Y, N_A, N_B)$

$AUTH_A := \mathrm{Sign}(SA_{A1}, X, N_A, N_B, \mathrm{prf}(ID_A))_{Sig_A^{KD}}$

$AUTH_B := \mathrm{Sign}(SA_{B1}, Y, N_B, N_A, \mathrm{prf}(ID_B))_{Sig_B^{KD}}$

Protocol (ESP)．A, B 間での ESP によるメッセージ通信のプロトコルを以下に示す．IKE により共通鍵 K_{shared} を A と B は共有しているものとする．共通鍵を用いて，メッセージ m_A または m_B を暗号化し，メッセージ認証コード（MAC）を生成し送信する．MAC の検証に成功すれば A と B の認証は済んだものとし，メッセージを受理する．

1. $A \to B$: $SPI_A, S_A, \mathrm{SK}\{m_A\}_{K_{shared}}, MAC_A$
2. $B \to A$: $SPI_B, S_B, \mathrm{SK}\{m_B\}_{K_{shared}}, MAC_B$

$MAC_A := \mathrm{Mac}(SPI_A, S_A, \mathrm{SK}\{m_A\}_{K_{shared}})_{K_{shared}}$

$MAC_B := \mathrm{Mac}(SPI_B, S_B, \mathrm{SK}\{m_B\}_{K_{shared}})_{K_{shared}}$

2.2 量子鍵配送

量子鍵配送（Quantum Key Distribution, QKD）は，量子物理による鍵配送の仕組みである．量子状態を伝送することができる量子通信路で，秘密鍵の情報を量子状態にエンコードして共有を行う．QKD の最大の特徴は，秘密鍵の共有時，第三者による盗聴（すなわち Passive Adversary）を検知できることである．これは「量子状態は測定すると状態が変化する」という量子系特有の性質に由来する．二者間で量子状態を送受信している途中で，第三者が盗聴，すなわち量子状態を測定すると，送信した量子状態と受信した量子状態が異なる場合がある．二者間で，それぞれ送受信した量子状態を公開された古典通信路で再送信し，それらが異なっている場合には第三者による盗聴が起きたと見なす．これらの量子状態の送受信と盗聴の検知を繰り返し行い，秘匿性の確保された鍵の割合を高めることで，最終的に二者間で秘密鍵を共有する．QKD には幾つかの方式があるが，標準化機関である ITU-T および ETSI で，実用化に向けた標準化が進められている [8, 9]．

QKD の Security Proof に関する文献 [7] においては，量子通信路と共に存在する古典通信路は認証されている必要があるとしており，本研究では後述するように固定した二

ⓒ 2021 Information Processing Society of Japan

拠点間の通信を想定して，QKD ベンダを信頼するモデル，すなわち古典通信路の認証は事前共有鍵相当としてモデルを整理した．

2.3 耐量子計算機暗号

耐量子計算機暗号 (Post-Quantum Cryptography, PQC) は，数学的に解くことが困難な問題を安全の根拠とする公開鍵暗号/鍵配送の仕組みであり，特に量子計算機でも難しいと予想される問題を安全性の根拠とする．例えば，格子暗号は格子点の集合が与えられたときに原点から最も近い格子点を求めることが量子計算機を用いても難しいと予想されていることを安全性の根拠としている．暗号に利用される数学的な問題の例としては，格子問題 (e.g. 最短ベクトル探索問題，最近ベクトル探索問題，LWE (Learning With Error) という機械学習に端を発する問題，LWE を環上で構成する Ring LWE 問題)，多変数多項式の求解に関する問題，伝送誤りのある通信路における符号の復号に関する問題，同種写像に関する問題などが候補とされている．米国標準技術研究所 (NIST) で現在進行形で PQC の標準化が進行しており，実用化が近付いている様子である [28]．

本論文では，PQC ベースの鍵交換を Post-Quantum cryptography-based Key Distribution (PQKD) と呼ぶ．また，QKD と PQKD を総称して xKD と呼ぶこととする．本論文で提案するセキュア光トランスポート NW において，xKD 装置は IPsec における IKE の役割を担う．

2.4 ホワイトボックススイッチ

提案するセキュア光トランスポート NW において，IPsec における ESP の役割を担うのがホワイトボックススイッチ (White-Boxed Switch, WBS) である [33]．WBS は，光トランスポートに伝送する情報を重畳するための光トランスポンダ機能を持ち，ホワイトボックストランスポンダと呼ばれることもある．従来，光トランスポートを担う伝送装置は，光モジュールや各種機能などが一体型で提供されていた．これに対し，ディスアグリゲーション構成と呼ばれる技術を採用している．これは，伝送装置の各種機能を分離し，標準化されたインタフェースで制御することで，柔軟な構成変更，付加機能の実現，コストの低減等が可能になる技術である．

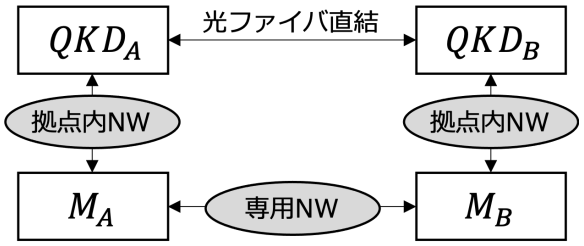
今回の提案では，伝送装置側に xKD 装置と連携する機能が必要となるため，WBS の採用が適する．xKD 装置と光トランスポンダを連携したセキュア光トランスポートの実証については，参考文献 [33] を参照されたい．

2.5 ProVerif

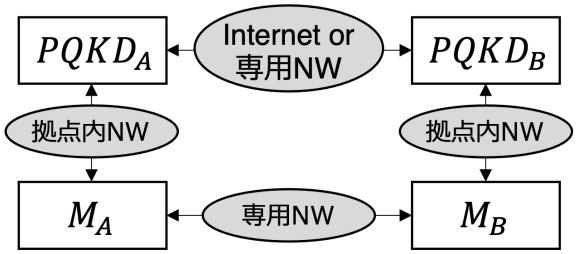
ProVerif [2,3] は暗号プロトコルの形式検証ツールであり，TLS 1.3 や 5G 認証プロトコルなど，様々なプロトコルの安全性検証に用いられている [1,31,32,34]．ProVerif では，Dolev-Yao モデル [6] と呼ばれる暗号プリミティブを理想化したモデルに対して，秘密情報の機密性や通信相手の認証などの安全性要件を検証することができる．検証コードの記述方法等，ProVerif の詳細については公式マニュアル [4] を参照されたい．

3. 問題設定

SOTN アーキテクチャでは，鍵交換は耐量子鍵配送方式と量子鍵配送方式の両者で検討を行う．図 2a に図示したように，量子鍵配送機器間は光ファイバ直結で接続されており，相互の認証も設置段階で完了している想定とする．一方，耐量子鍵配送方式においては図 2b のように通常のインターネット回線を用いた鍵配送が想定される．上記いずれかの方式において交換した共通鍵を用いて暗号化されたデータを，光トランスポート NW を用いた大容量の通信を行えるメッセージ配送機器を用いて伝送する．



(a) 量子鍵配送を用いる場合.



(b) 耐量子鍵配送を用いる場合.

図 2: SOTN アーキテクチャの模式図．拠点 X の量子鍵配送装置を QKD_X，耐量子鍵配送装置を $PQKD_X$，メッセージ配送装置を M_X とする．

量子/耐量子鍵配送機器と光トランスポートをサポートしたメッセージ配送機器は，ディスアグリゲーション構成により鍵交換と伝送が別機器となることを想定する．鍵配送機器とメッセージ配送機器はそれぞれの拠点の拠点内 NW によって接続されるものとする．

SOTN アーキテクチャの暗号プロトコル設計にあたり，拠点間通信のデファクトスタンダードである IPsec をベースに考える．IPsec においては，IKE に従う鍵交換と ESP に従うメッセージ配送を同一のネットワーク機器で実行することを想定している．一方で，SOTN アーキテクチャで

は鍵交換機器とメッセージ配送機器が分離することから，IKE と ESP は別機器で実行する必要がある．よって，鍵交換機器とメッセージ配送機器間の認証が必要となる．

また，鍵交換プロトコルの安全性を考える上で，重要な要件が 2 つある．1 つ目は Perfect Forward Secrecy (PFS) である．PFS とは，パーティの長期的な秘密情報（例えば公開鍵に紐づく秘密鍵や署名鍵など）があるタイミングで漏洩したとしても，そのタイミング以前の通信内容の機密性が保たれるという安全性であり，スノーデン事件などを受けて重要性が高まっている．また，2 つ目は Key Compromise Impersonation (KCI) 攻撃への耐性である．KCI 攻撃とは，あるパーティの長期的な秘密情報が漏洩したときに，攻撃者がそのパーティ以外のパーティになりすます攻撃である．この攻撃が可能であると，悪意のあるパーティが他人になりすますことができたり，一人のパーティの長期秘密鍵が漏洩すると攻撃者が任意のパーティになりすますことができたりしてしまう．（あるパーティの長期秘密情報が攻撃者に漏洩したとき，攻撃者がそのパーティ自身になりすますことができるのは明らかであるため，そのような場合は安全性要件となり得ない.）上記の安全性要件をとらえるために，SOTN アーキテクチャの攻撃者モデルではパーティの署名鍵の漏洩を考える．

以上のことから，SOTN アーキテクチャの攻撃者モデルを以下のように設定する．攻撃者は各鍵交換機器間，各メッセージ配送機器間の通信路を完全に支配する．すなわち，攻撃者は通信路を流れるメッセージの盗聴，改ざん，停止，再送が可能である．ただし，量子鍵配送方式を用いる通信路（光ファイバ直結により構成されている通信路）に対しては，攻撃者は上記のいずれの行為も行うことができない．さらに，攻撃者はパーティの長期秘密鍵（署名鍵）を手に入れることができる．

本論文では，上記の攻撃者に対し 4.2 節で述べる安全性要件が満たされているかどうかを検証する．

4. 提案プロトコルと安全性要件

4.1 提案手法

拠点 A と拠点 B における鍵配送及びメッセージ配送を考える．以降で 3 点のプロトコルを提案する．提案プロトコル 1 は耐量子鍵配送を用いた SOTN の暗号プロトコルであり，提案プロトコル 2 は提案プロトコル 1 の鍵配送を量子鍵配送で行うものである．提案プロトコル 3 は提案プロトコル 2 を改良したものである．以降，拠点 X の耐量子鍵交換機器及び量子鍵交換機器を KD_X，メッセージ配送機器を M_X とおく．

4.1.1 提案手法 1．PQKD-SignedDH

耐量子鍵配送を用いて KD_A-KD_B 間で共通鍵 K_{shared} を共有することを考える．また，KD_X-M_X 間（$X \in \{A, B\}$）では署名付き DH 鍵配送プロトコル（SignedDH）を用いて，K_{shared} を配送する．

Protocol (PQKD-SignedDH)**.** あらかじめ，KD_A の検証鍵 Ver_A^{KD} と署名鍵 Sig_A^{KD}，KD_B の検証鍵 Ver_B^{KD} と署名鍵 Sig_B^{KD} および，M_A の検証鍵 Ver_A^{OTN} と署名鍵 Sig_A^{OTN}，M_B の検証鍵 Ver_B^{OTN} と署名鍵 Sig_B^{OTN} を発行しておく．

まず，暗号プリミティブを PQC とした IKE プロトコル（IKE-PQC と呼ぶ）を適用し，共通鍵 K_{Shared} を交換する．この際，認証においては Ver_A^{KD} と Sig_A^{KD}，Ver_B^{KD} と Sig_B^{KD} を使用する．

次に，KD_A-M_A 間で共通鍵 K_{Shared} を配送する．これには SignedDH を用いる．この際には Sig_A^{KD}, Ver_A^{KD}, Ver_A^{OTN}，Sig_A^{OTN} を認証に使用する．

同様に拠点 B 側でも SignedDH により，共通鍵 K_{Shared} をメッセージ配送機器に送る．これにより拠点 A と拠点 B の間のメッセージ配送を ESP プロトコルを用いて実行できる．なお，メッセージ配送機器同士の認証は ESP に定められている通り，交換した共通鍵を用いた共通鍵認証である．

1. $KD_A \leftrightarrow KD_B$: IKE − PQC
2. $KD_A \rightarrow M_A$: SignedDH
3. $KD_B \rightarrow M_B$: SignedDH
4. $M_A \leftrightarrow M_B$: ESP

以下に SignedDH プロトコルを示す．

Protocol (SignedDH)**.** G を素位数 p の有限可換群とし，G の生成元を $g \in G$ とする．KD_X と M_X はそれぞれ，$x \xleftarrow{U} \mathbb{Z}_p$，$y \xleftarrow{U} \mathbb{Z}_p$ を選ぶ．以下の手順に従い，KD_X から M_X に鍵を配送する．

1. $KD_X \rightarrow M_X$: $g^x, \mathrm{Sign}(g^x)_{Sig_X^{KD}}$
2. $M_X \rightarrow KD_X$: $g^y, \mathrm{Sign}(g^y)_{Sig_X^{OTN}}$
3. $KD_X \rightarrow M_X$: $\mathrm{SK}\{K_{shared}\}_{g^{xy}}$

4.1.2 提案手法 2．QKD-SignedDH

量子鍵配送（QKD）を用いて KD_A-KD_B 間で共通鍵 K_{shared} を共有することを考える．また，KD_X-M_X 間（$X \in \{A, B\}$）では SignedDH を用いて K_{shared} を配送する．

Protocol (QKD-SignedDH)**.** まず量子鍵配送（QKD）を用いて共通鍵 K_{Shared} を交換する．この際，認証は量子鍵配送機器の性質上，事前に完了している想定とする．次に，提案手法 1 と同様に各拠点で KD_X と M_X 間で共通鍵 K_{Shared} を SignedDH に従い配送する．

1. $KD_A \leftrightarrow KD_B$: QKD
2. $KD_A \rightarrow M_A$: SignedDH
3. $KD_B \rightarrow M_B$: SignedDH
4. $M_A \leftrightarrow M_B$: ESP

Vol.2021-CSEC-95 No.19
Vol.2021-SPT-45 No.19
Vol.2021-EIP-94 No.19
2021/11/9

4.1.3 提案手法 3．QKD-SignedDH-Sign

提案プロトコルをベースに，M_A-M_B 間の認証を強化したプロトコルを考える．

Protocol (QKD-SignedDH-Sign)．まず量子鍵配送（QKD）によって共通鍵 K_{Shared} を交換する．次に，提案手法 1 と同様に各拠点で KD_X と M_X の間で共通鍵 K_{Shared} を SignedDH に従い配送する．その後，ESP に定められた共通鍵認証に加えて，M_A-M_B 間で相互証明書認証付きのメッセージ配送を行う．すなわち，メッセージ配送時に署名検証鍵 Sig_A^{OTN}, Ver_A^{OTN} および Sig_B^{OTN}, Ver_B^{OTN} を用いて検証を行う．

1. $KD_A \leftrightarrow KD_B$: QKD
2. $KD_A \rightarrow M_A$: SignedDH
3. $KD_B \rightarrow M_B$: SignedDH
4. $M_A \rightarrow M_B$: $SPI_A, S_A, \mathrm{SK}\{m_A\}_{K_{shared}}, MAC_A, ah_A$
5. $M_B \rightarrow M_A$: $SPI_B, S_B, \mathrm{SK}\{m_B\}_{K_{shared}}, MAC_B, ah_B$

$ah_A \coloneqq \mathrm{Sign}(SPI_A, S_A, \mathrm{SK}\{m_A\}_{K_{shared}}, MAC_A)_{Sig_A^{OTN}}$

$ah_B \coloneqq \mathrm{Sign}(SPI_B, S_B, \mathrm{SK}\{m_A\}_{K_{shared}}, MAC_B)_{Sig_B^{OTN}}$

4.2 安全性要件

本節では，今回検証する安全性要件を述べる．なお，Key Compromise Disagreement for Multi-party (KCD) は本研究で新しく定義した安全性要件である．これは，二者間のプロトコルの安全性要件である KCI 攻撃への耐性を三者間以上のプロトコルにも適用できるように拡張した安全性要件である．本研究で提案するプロトコルは四者間のプロトコルのため，KCD を検証の対象とする．

Agreement．拠点 A の鍵交換機器とメッセージ配送機器，及び拠点 B の鍵交換機器とメッセージ配送機器の 4 点のエンティティ間の合意要件．図 3 の A1-A8 の 8 要件を考える．なお，合意とは 2 エンティティ間において，通信後，一方のエンティティが他方のエンティティが取得していていることを期待する値を他方が正しく所持しているという要件である．

Secrecy．通信路を流れるメッセージの内容が攻撃者に知られないという要件．

Perfect Forward Secrecy（PFS）．たとえ署名鍵が漏洩したとしても，漏洩以前のセッションのメッセージの内容は攻撃者に知られないという要件．

Key Compromise Impersonation（KCI）．二者間のプロトコルにおいて，一方の署名鍵が漏洩したときに，攻撃者は他方になりすまし合意を破ることができるか（例えば，Sig_A が漏洩したときに攻撃者は B になりすますことができるか）という要件．

Key Compromise Disagreement for Multi-party (KCD)．署名鍵が漏洩したときに各エンティティ間の合意が成り立つかという要件．KCI の拡張になっている．

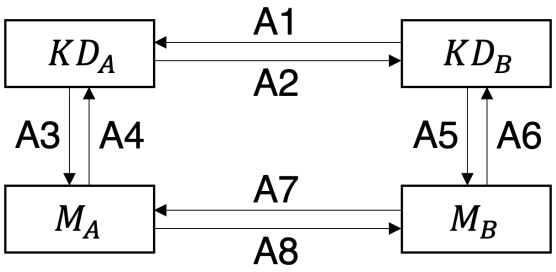


図 3: Agreement 要件の図．例えば，A1 は KD_B から KD_A への合意を表す．

5. 評価

提案手法 1, 2, 3 の Proverif による検証を行う．この検証により検知された攻撃の詳細を延べ，本評価により提案手法 3 が上記攻撃への対策となっていることを示す．

5.1 安全性検証

提案手法 1, 2, 3 について Proverif による検証を行った．Secrecy, PFS, KCI についての検証結果を表 1 に示す．いずれについても True（攻撃発見なし）であり，安全であることが確認できた．なお，KCI(Sig) は署名鍵 Sig の漏洩に対する KCI 要件を表す．

表 1: Secrecy，PFS，KCI の検証結果

	提案手法 1	提案手法 2	提案手法 3
Secrecy	True	True	True
PFS	True	True	True
KCI(Sig_A^{KD})	True	True	True
KCI(Sig_B^{KD})	True	True	True
KCI(Sig_A^{OTN})	True	True	True
KCI(Sig_B^{OTN})	True	True	True

次に，表 2，表 3，表 4 に Agreement の検証結果を示す．A1-A8 は図 3 に表される合意を表し，Agreement 列はその合意要件が満たされているかを表す．表 2 においては，KCD(Sig_A^{KD})-(A1, A2, A3, A4) をまとめたものが KCI(Sig_A^{KD}) であり，KCD(Sig_B^{KD})-(A1, A2, A5, A6) をまとめたものが KCI(Sig_B^{KD}) である．表 3 と表 4 においては，KCD(Sig_A^{KD})-(A3, A4) をまとめたものが KCI(Sig_A^{KD}) であり，KCD(Sig_B^{KD})-(A5, A6) をまとめたものが KCI(Sig_B^{KD}) である．

本検証により，提案手法 1, 2, 3 においてすべてのエンティティ間の合意要件が満たされていることが確認できた．しかしながら，KCD(Sig_A^{KD})，および KCD(Sig_B^{KD}) においては False（攻撃発見）がいくつか見受けられる．

ⓒ 2021 Information Processing Society of Japan

表 2: 提案手法 1 の Agreement と KCD 要件の検証結果

	Agreement	KCD(Sig_A^{KD})	KCD(Sig_B^{KD})
A1	True	True	False
A2	True	False	True
A3	True	False	True
A4	True	True	True
A5	True	True	False
A6	True	True	True
A7	True	False	False
A8	True	False	False

表 3: 提案手法 2 の Agreement と KCD 要件の検証結果

	Agreement	KCD(Sig_A^{KD})	KCD(Sig_B^{KD})
A1	True	True	True
A2	True	True	True
A3	True	False	True
A4	True	True	True
A5	True	True	False
A6	True	True	True
A7	True	False	True
A8	True	True	False

表 4: 提案手法 3 の Agreement と KCD 要件の検証結果

	Agreement	KC(Sig_A^{KD})	KC(Sig_B^{KD})
A1	True	True	True
A2	True	True	True
A3	True	False	True
A4	True	True	True
A5	True	True	False
A6	True	True	True
A7	True	True	True
A8	True	True	True

まず表 2 について考察する．本表で網掛けの部分は対応する KCI 要件を表している．KCD(Sig_A^{KD})-(A2, A3) および KCD(Sig_A^{KD})-(A1, A5) の False は自明なりすましであるため，一般的な鍵交換プロトコルの安全性要件の検討においては許容される．

一方，KCD(Sig_A^{KD})-(A7, A8)，および，KCD(Sig_B^{KD})-(A7, A8) の False については，マルチパーティ特有の攻撃が検知された．これらの攻撃については次節で詳細を述べる．また，表 3 の提案手法 2 においても KCD(Sig_A^{KD})-A7 および KCD(Sig_B^{KD})-A8 の脆弱性が残っている．これについても検知した攻撃を次節で述べる．一方で，表 4 から，提案手法 3 では提案手法 1, 2 で見られた脆弱性が解消されていることがわかる．

5.1.1 ProVerif による検証で検出された攻撃

提案手法 1 において，署名鍵 Sig_A^{KD} が漏洩した場合に検知された攻撃について述べる．これは表 2 の KCD(Sig_A^{KD})-(A7, A8) の False に対応する．攻撃者は Sig_A^{KD} を用いて，拠点 A の鍵配送機器になりすまし，拠点 A のメッセージ配送機器との間の認証，および拠点 B の鍵配送機器との間の認証を突破することができる．これにより，攻撃者は偽の共通鍵 $\bar{K}_{Shared}$ を拠点 A のメッセージ配送機器と拠点 B の鍵配送機器に送信することができる．拠点 B の鍵配送機器は，プロトコルに従って $\bar{K}_{Shared}$ を正しい共通鍵だと思いこみ，拠点 B のメッセージ配送機器にこれを配送してしまう．以上により，メッセージ配送時の共通鍵は攻撃者の知るところとなっているため，メッセージ配送機器間の合意 A7, A8 を破り，メッセージを取得することができる．署名鍵 Sig_B^{KD} が漏洩した場合においても同様であり，これは表 2 の KCD(Sig_B^{KD})-(A7, A8) の False に対応する．

以下にその攻撃アルゴリズムを示す．

1. $Attacker \leftrightarrow KD_B$: IKE – PQC
2. $Attacker \rightarrow M_A$: SignedDH
3. $KD_B \rightarrow M_B$: SignedDH
4. $Attacker \rightarrow M_B$: ESP
5. $M_A \leftarrow Attacker$: ESP

次に，提案手法 2 において，署名鍵 Sig_A^{KD} が漏洩した場合に検知された攻撃について述べる．これは表 3 の KCD(Sig_A^{KD})-A7 の False に対応する．攻撃者は Sig_A^{KD} を用いて拠点 A の鍵配送機器になりすまし，拠点 A のメッセージ配送機器に偽の共通鍵 $\bar{K}_{Shared}$ を送信することができる．攻撃者は $\bar{K}_{Shared}$ を所持しているので，これを用いてメッセージ配送機器 B に成りすまし，共通鍵認証を突破し，合意 A7 を破り，メッセージを傍受することができる．

以下にその攻撃アルゴリズムを，図 4 に ProVerif によって出力された攻撃トレースを示す．

1. $KD_A \leftrightarrow KD_B$: QKD
2. $Attacker \rightarrow M_A$: SignedDH
3. $KD_B \rightarrow M_B$: SignedDH
4. $Attacker \rightarrow M_B$: ESP

備考．提案手法 3 の鍵交換を耐量子鍵配送で行うプロトコル（PQKD-SignedDH-Sign）では，ここで紹介した攻撃を防ぐことはできないことが確かめられている．

5.1.2 ProVerif による検証で得られた知見

まず，本研究において，KCI の拡張である KCD という安全性要件を新たに提案したことで，5.1.1 で表されるような複数エンティティ間でのなりすましに対する攻撃を捉えることができた．

表 2 と表 3 を比較すると，提案手法 1 では満たされていなかった KCD(Sig_A^{KD})-A8 および KCD(Sig_B^{KD})-A7 の安全性要件が提案手法 2 では満たされていることがわかる．すなわち鍵配送に Passive Advarsary を許さない通信路を

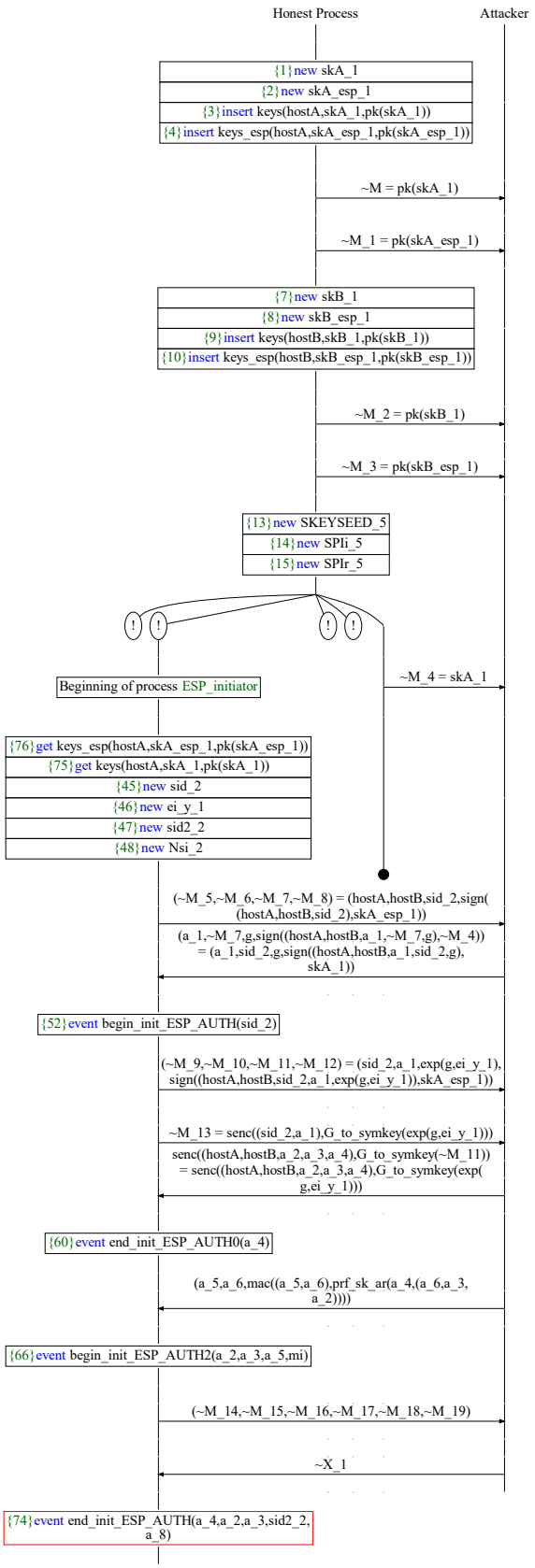


図 4: 提案手法 2 で合意 A7 を破る KCD(Sig_A^{KD}) Attack を表す Proverif の出力.

用いることで，KCD Attack に対する耐性が向上することがわかった．これは量子鍵配送の優位性を示すものである．提案手法 3 は提案手法 2 にさらにメッセージ配送機器間の相互認証を加えて，共通鍵認証とのハイブリッド認証とすることで，KCD Attack を完全に防ぐことができることを示した．

6. 関連研究

本節では，提案プロトコルのベースとなった IPsec の形式検証に関する既存研究を紹介する．なお，既存研究はいずれも IKE のみを検証対象としている．

1999 年，Meadows によって IKE プロトコルの安全性検証に初めて形式検証技術が用いられた．Meadows は NRL プロトコルアナライザーを用いて IKEv1 の安全性検証を行い，リフレクション攻撃の可能性と認証要件に対する脆弱性を発見した [18]．Cremers は暗号プロトコルのモデル検査ツールである Scyther を用いて IKE の包括的な検証を行った [5]．その中でそれまでに報告されていなかった脆弱性を発見し，強い認証要件が常に満たされるわけではないことを示した．また，IKEv1 で攻撃の可能性が指摘されていたリフレクション攻撃が IKEv2 でも可能であることを発見した．Küsters らは ProVerif を DH 鍵交換にも使えるように拡張し，その中で IKE の安全性検証を行った [17]．そして今までに発見されていた脆弱性をとらえられることを確認した．Ninet らはモデル検査ツール Spin を用いて IKEv2 プロトコルを検証し，リフレクション攻撃が現実的ではないこと，認証要件に対する脆弱性が存在することを示した [27]．また，NICT による暗号プロトコルの安全性評価ポータルサイト Cryptographic Protocol Verification Portal (CPVP) [26] では，IKE の ProVerif や Scyther による安全性評価結果 [19–25] が掲載されている．

7. まとめと今後の課題

本研究では量子コンピュータに対して安全な光トランスポート NW である，SOTN アーキテクチャに対する安全な通信プロトコルを検討した．SOTN アーキテクチャは従来の拠点間通信のデファクトスタンダードである IPsec が想定している機器構成と異なっていることを述べ，IPsec ベースの新たな通信プロトコルを提案した．さらに KCD という新たな要件を含む安全性要件を定義し，これに則って ProVerif を用いた安全性検証を行った．

今後の課題としては，提案プロトコルについて，ラウンド数，処理時間などの観点でのプロトコルのトレードオフの調査や，鍵交換機器とメッセージ配送機器間の鍵配送を事前共有鍵方式で行った場合の検証，さらに，メッセージ配送機器側の署名鍵が漏洩した場合の KCD 要件を検証することなどが挙げられる．また，本研究では QKD の認証について，QKD 装置のベンダを信頼する前提で，事前共有鍵を安全に装置間で共有できることを前提とした．今後，QKD 装置のユースケースとして，固定した拠点間のみで

ⓒ 2021 Information Processing Society of Japan

なくネットワーク化を想定する場合には，任意の 2 地点を接続するために，公開鍵ベース，特に PQC ベースの認証を組み合わせることが妥当だと考えられる [8]．また，現状 QKD 装置は “Trusted Node” として扱われるが [10]，現在 device-independent，すなわち “Trusted Node” を想定しない QKD が理論的に検討されており，将来的には “Trusted Node” を想定しない xKD 鍵交換装置の採用が望ましいと考えられる．これら関連分野の研究開発動向については，引き続き注視すべきである．

参考文献

[1] Bhargavan, K., Blanchet, B. and Kobeissi, N.: Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate, *IEEE Symposium on Security and Privacy (S&P'17)*, San Jose, CA, IEEE, pp. 483–503 (2017). Distinguished paper award.

[2] Blanchet, B.: Automatic verification of security protocols in the symbolic model: The verifier proverif, *Foundations of security analysis and design VII*, Springer, pp. 54–87 (2013).

[3] Blanchet, B.: Modeling and verifying security protocols with the applied pi calculus and ProVerif, *Foundations and Trends® in Privacy and Security*, Vol. 1, No. 1-2, pp. 1–135 (2016).

[4] Blanchet, B., Smyth, B., Cheval, V. and Sylvestre, M.: Proverif 2.00: Automatic cryptographic protocol verifier, user manual and tutorial (2020).

[5] Cremers, C.: Key exchange in IPsec revisited: Formal analysis of IKEv1 and IKEv2, *European Symposium on Research in Computer Security*, Springer, pp. 315–334 (2011).

[6] Dolev, D. and Yao, A.: On the security of public key protocols, *IEEE Transactions on information theory*, Vol. 29, No. 2, pp. 198–208 (1983).

[7] ETSI: Quantum Key Distribution (QKD); Security Proofs (V1.1.1), *GS QKD*, No. 005 (2010).

[8] ETSI: Quantum Safe Cryptography and Security, *ETSI White Paper*, No. 8 (2015).

[9] ETSI: Implementation Security of Quantum Cryptography (First edition), *ETSI White Paper*, No. 27 (2018).

[10] ETSI: Quantum Key Distribution (QKD); Protocol and data format of REST-based key delivery API (V1.1.1), *GS QKD*, No. 014 (2019).

[11] IEEE: 802.1AE Media Access Control (MAC) Security.

[12] IEEE: 802.1X Local and Metropolitan Area Networks - Port-Based Network Access Control.

[13] Kaufman, C., Hoffman, P. E., Nir, Y., Eronen, P. and Kivinen, T.: Internet Key Exchange Protocol Version 2 (IKEv2), *RFC*, Vol. 7296, pp. 1–142 (2014).

[14] Kent, S. T.: IP Authentication Header, *RFC*, Vol. 4302, pp. 1–34 (2005).

[15] Kent, S. T.: IP Encapsulating Security Payload (ESP), *RFC*, Vol. 4303, pp. 1–44 (2005).

[16] Kent, S. T. and Seo, K.: Security Architecture for the Internet Protocol, *RFC*, Vol. 4301, pp. 1–101 (2005).

[17] Küsters, R. and Truderung, T.: Using ProVerif to analyze protocols with Diffie-Hellman exponentiation, *2009 22nd IEEE Computer Security Foundations Symposium*, IEEE, pp. 157–171 (2009).

[18] Meadows, C.: Analysis of the Internet Key Exchange protocol using the NRL protocol analyzer, *Proceedings of the 1999 IEEE Symposium on Security and Privacy (Cat. No. 99CB36344)*, IEEE, pp. 216–231 (1999).

[19] NICT: IKE-PKE の Scyhter による評価結果. CPVP 技術文書, `https://crypto-protocol.nict.go.jp/AKE_zoo/IKE-PKE/IKE-PKE_Scyther.pdf`.

[20] NICT: IKE-PSK の ProVerif による評価結果. CPVP 技術文書, `https://crypto-protocol.nict.go.jp/AKE_zoo/IKE-PSK/IKE-PSK_ProVerif.pdf`.

[21] NICT: IKE-PSK の Scyhter による評価結果. CPVP 技術文書, `https://crypto-protocol.nict.go.jp/AKE_zoo/IKE-PSK/IKE-PSK_Scyther.pdf`.

[22] NICT: IKE-SIG の ProVerif による評価結果. CPVP 技術文書, `https://crypto-protocol.nict.go.jp/AKE_zoo/IKE-SIG/IKE-SIG_ProVerif.pdf`.

[23] NICT: IKE-SIG の Scyhter による評価結果. CPVP 技術文書, `https://crypto-protocol.nict.go.jp/AKE_zoo/IKE-SIG/IKE-SIG_Scyther.pdf`.

[24] NICT: IKEv2 の ProVerif による評価結果. CPVP 技術文書, `https://crypto-protocol.nict.go.jp/AKE_zoo/IKEv2/IKEv2_ProVerif.pdf`.

[25] NICT: IKEv2 の Scyhter による評価結果. CPVP 技術文書, `https://crypto-protocol.nict.go.jp/AKE_zoo/IKEv2/IKEv2_Scyther.pdf`.

[26] NICT: Cryptographic Protocol Verification Portal (2021). `https://crypto-protocol.nict.go.jp/`. Accessed: 2021-10-15.

[27] Ninet, T., Legay, A., Maillard, R., Traonouez, L.-M. and Zendra, O.: Model checking the IKEv2 protocol using Spin, *2019 17th International Conference on Privacy, Security and Trust (PST)*, IEEE, pp. 1–7 (2019).

[28] NIST: Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process, *NISTIR*, Vol. 8309 (2020).

[29] Xie, C., Wang, L., Dou, L., Xia, M., Chen, S., Zhang, H., Sun, Z. and Cheng, J.: Open and disaggregated optical transport networks for data center interconnects, *IEEE/OSA Journal of Optical Communications and Networking*, Vol. 12, No. 6 (2020).

[30] Yamazaki, E., Tomizawa, M. and Miyamoto, Y.: 100-Gb/s optical transport network and beyond employing digital signal processing, *IEEE Commun. Mag.*, Vol. 50, No. 2 (2012).

[31] Zhang, J., Yang, L., Cao, W. and Wang, Q.: Formal Analysis of 5G EAP-TLS Authentication Protocol Using Proverif, *IEEE Access*, Vol. 8, pp. 23674–23688 (online), DOI: 10.1109/ACCESS.2020.2969474 (2020).

[32] Zhang, J., Yang, L., Gao, X., Tang, G., Zhang, J. and Wang, Q.: Formal Analysis of QUIC Handshake Protocol Using Symbolic Model Checking, *IEEE Access*, Vol. 9, pp. 14836–14848 (online), DOI: 10.1109/ACCESS.2021.3052578 (2021).

[33] 奥田哲矢，千田浩司，白井大介，知加良盛，齋藤恆和，中林美郷，山村和輝，田中友里，夏川勝行，高杉耕一：セキュア光トランスポートネットワーク，NTT 技術ジャーナル，Vol. 11 (2021).

[34] 荒井研一，　渡辺大，櫻田英樹ほか：ProVerif による TLS1. 3 ハンドシェイクプロトコルの形式検証，コンピュータセキュリティシンポジウム 2015 論文集，Vol. 2015, No. 3, pp. 1003–1010 (2015).

ⓒ 2021 Information Processing Society of Japan