実環境を想定したWPA2に対するKRACKsの評価実験

窪田 恵人[1,a)]　小家 武[2]　船引 悠生[2]　藤堂 洋介[3]　五十部 孝典[4]　森井 昌克[2]

概要：2017 年に “KRACKs” と呼ばれる Wi-Fi の暗号化方式に対する攻撃が提案された．KRACKs はハンドシェイクの脆弱性を利用した攻撃であり，特に WPA2 に対する影響が懸念されている．しかし，KRACKs の実環境における影響を評価する研究はまだ少ない．そこで本稿では，実際にいくつかの OS に対して KRACKs を行うことでその影響を調べた．さらに KRACKs の前提となる中間者攻撃についても，どういった環境で実行可能であるのかを調査した．結果として，中間者攻撃は攻撃者が被害者に近づくことさえできれば，障害物の有無に関係なく攻撃が成功することが分かった．KRACKs に関しては，いくつかのパターンで KRACKs が対策される前の環境では攻撃に部分的に成功したが，パッチによって対策された後の環境では失敗した．また，先行研究とは違った挙動も観測したのでこれを報告する．

キーワード：Wi-Fi，WPA2，KRACKs，中間者攻撃

1. はじめに

ホテルやカフェなどの商業施設，図書館や空港といった公共施設などでも，無線 LAN によるインターネット接続，Wi-Fi を利用する機会が増えている．特に近年は働き方改革の推進を背景に，リモートワークを実現する手段として，ますますその需要が高まっている．しかしながら，データの送受信に電波を使用する Wi-Fi は盗聴が容易であり，安全なデータのやり取りには通信の暗号化が必要不可欠である．現在，Wi-Fi の暗号化方式として最も広く用いられているものが WPA2 と呼ばれる方式である．

2017 年，WPA2 に対して KRACKs（Key Reinstallation AttaCKs）[1] と呼ばれる攻撃が提案された．この攻撃は WPA2 のセキュリティプロトコルの脆弱性を利用するものである．攻撃者は特定のパケットを遮断することで，カウンターモードの暗号化に用いるカウンター値を強制的にリセットさせることができるとされる．カウンター値がリセットされると攻撃者はパケットを復号するためのヒントを得ることができ，通信内容が解析されてしまう恐れがある．また，パケットの遮断のためにはまず中間者攻撃を行う必要があるが，その一つとしてチャネルベース中間者攻撃が存在する [2]．この攻撃に成功することで KRACKs が可能となるため，チャネルベース中間者攻撃を KRACKs の前提条件として考えることができる．

KRACKs がセキュリティ業界に大きな波紋を広げたことは確かであるが，その内容や及ぼす影響を正確に把握している者は少ないように思われる．正しく KRACKs を理解し，Wi-Fi を安全に使用していくためにも，実際にその影響を評価していくことが重要である．

そこで我々は，KRACKs の現実的な脅威度を評価するため，実環境を想定した実験を行い，その影響について調査を行った．具体的には，様々な環境でチャネルベース中間者攻撃を行い，中間者攻撃と KRACKs の実現可能性を調べた．また，KRACKs の影響を調べるためにいくつかの OS に対してパケットの復号を試みた．

結果として，アクセスポイントからの電波が届く範囲であれば，攻撃者は物理的にアクセスポイントとクライアントの間に入らなくともチャネルベース中間者攻撃に成功することが分かった．また一部の OS では，KRACKs に未対策の状態でパケットの復号ができることを確認した．ただし，KRACKs が提案された後，主要な OS では 2017 年 10 月から 11 月にパッチによる対策がなされており，パッチ適用後の環境では攻撃に失敗したため，適切なアップデートを行っていれば現在では脅威とはならない．さらに，KRACKs を行っても暗号化に用いられるカウンター値がリセットされない，攻撃後に通信が続かず接続が切れてしまうため通信を傍受し続けることができないといった KRACKs が提案された論文では述べられていない挙動を二点観測した．

[1] 神戸大学工学部
[2] 神戸大学大学院工学研究科
[3] NTT セキュアプラットフォーム研究所
[4] 兵庫県立大学大学院応用情報科学研究科
[a)] kubota@stu.kobe-u.ac.jp

2. WPA2 とそれに対する攻撃

2.1 WPA2

WPA2（Wi-Fi Protected Access 2）は無線 LAN のセキュリティについて定めた国際標準である IEEE 802.11i [3] を元に策定された技術規格の一つであり，暗号化プロトコルとして AES（Advanced Encryption Standard）を用いた CCMP（Counter mode with CBC-MAC Protocol）を採用している．ネットワークへの接続を要求するクライアントとそれを認証するアクセスポイントとの間で以下に示す三段階の処理が行われる．

一段階目の処理はアソシエーション処理と呼ばれ，クライアントとアクセスポイントの間でオープンシステム認証（IEEE 802.11 認証 [4]）が行われる．

二段階目の処理では，アクセスポイントとクライアントがマスター鍵（Pairwise Master Key）を共有する．PSK（Pre-Shared Key）認証と IEEE 802.1X 認証 [5] という 2 種類の認証方式が用意されている．

三段階目の処理では，4 ウェイハンドシェイクと呼ばれるプロトコルを用いてユニキャスト通信のためのペア暗号鍵（Pairwise Transient Key），グループ鍵ハンドシェイクを用いてマルチ/ブロードキャスト通信のためのグループ鍵（Group Transient Key）を生成/共有する．

また，クライアントが移動するときに接続を途切れさせずに素早くアクセスポイントを切り替えるローミングを行う際は一段階目の後に高速移行（Fast BSS Transition，FT）ハンドシェイク [6] によってペア暗号鍵とグループ鍵の生成/共有を行う．

WPA2 ではこれらの手順で生成/共有された暗号鍵を用いて暗号通信を行う．また，KRACKs はこれら 3 つのハンドシェイクに対する攻撃であるが，実環境を想定した場合に最も使用率が高く，かつ重要な通信が行われる可能性が高いのはペア暗号鍵を用いたユニキャスト通信であると考え，本稿では 4 ウェイハンドシェイクに対する攻撃を検討した．

2.1.1 4 ウェイハンドシェイク

4 ウェイハンドシェイクは通信データの暗号化に用いるペア暗号鍵をアクセスポイント（Authenticator）とクライアント（Supplicant）の両方で生成/共有するためのプロトコルである．ペア暗号鍵はアクセスポイントのパスフレーズと MAC アドレスから事前に計算されるマスター鍵および双方の MAC アドレスと二つの乱数（ANonce，SNonce）から生成される．また，4 ウェイハンドシェイクではグループ鍵がアクセスポイントからクライアントに一方的に送信される．図 1 で 4 ウェイハンドシェイクの流れを示した．

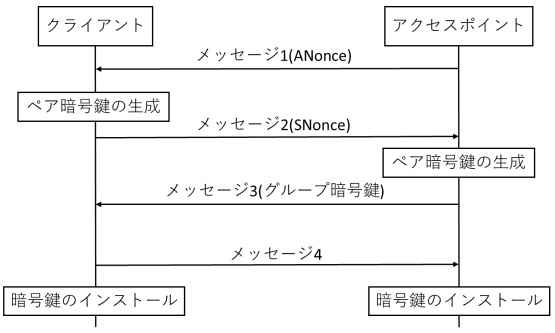

図 1 4 ウェイハンドシェイクのプロセス

メッセージ 1（アクセスポイント → クライアント）
アクセスポイントは乱数 ANonce を生成し，クライアントに送信する．

メッセージ 2（クライアント → アクセスポイント）
クライアントは乱数 SNonce を生成し，アクセスポイントに送信する．ここでクライアントはペア暗号鍵の生成に必要な情報をすべて手に入れたのでペア暗号鍵を生成する．

メッセージ 3（アクセスポイント → クライアント）
アクセスポイントはメッセージ 2 を受け取ることでペア暗号鍵の生成に必要な情報をすべて手に入れるのでペア暗号鍵を生成する．生成したペア暗号鍵でグループ鍵を暗号化してクライアントに送信する．

メッセージ 4（クライアント → アクセスポイント）
クライアントは受け取ったメッセージ 3 を復号してグループ鍵を確認する．その後，クライアントは確認応答としてメッセージ 4 を送信しペア暗号鍵とグループ鍵をインストールする．アクセスポイントはメッセージ 4 を受信した後にペア暗号鍵をインストールする．

2.1.2 暗号化プロトコル

WPA2 の機密性と完全性を保証するセキュリティプロトコルは AES-CCMP と呼ばれる．これは AES 暗号をカウンターモード（counter mode with CBC-MAC）で利用する方式であり，認証付き暗号（Authenticated Encryption with Associated Data, AEAD）かつ，ある特定の暗号化鍵の下で初期化ベクトルが繰り返されない限りは安全であることが知られている．

WPA2 では，送信者の MAC アドレスや各パケットに割り振られるパケット番号などを組み合わせて，この初期化ベクトルを生成する．パケット番号は，アクセスポイントおよびクライアントがパケットを送信するたびにインクリメントされる．また，同じペア暗号鍵で初期化ベクトルを再利用しないように，鍵がインストールされるたびにパケット番号をリセットする設計がなされている．仮に同じペア暗号鍵で初期化ベクトルが再利用された場合，同じ初期化ベクトルで暗号化された 2 つの暗号文の排他的論理和とそれに対応する平文の排他的論理和が等しくなるため平文に関する情報が漏洩してしまう．

2.2 チャネルベース中間者攻撃

チャネルベース中間者攻撃は中間者攻撃の一種として，2014 年に Vanhoef らによって提案された [2]．

クライアントがアクセスポイントに接続するためには周囲にあるアクセスポイントの情報を収集する必要がある．そのために，ビーコンおよびプローブ要求が用いられる．ビーコンとは，アクセスポイントがクライアントに対して自身の SSID（Service Set IDentifier）や通信方式などを通知する信号のことである．プローブ要求ではクライアントがすべてのアクセスポイント宛に調査を行い，プローブ要求を受け取ったアクセスポイントはプローブ応答として自身の SSID などを通知する．

チャネルベース中間者攻撃において攻撃者は，攻撃対象となる正規のアクセスポイントのチャネル（周波数）を妨害することでクライアントが正規のアクセスポイントの情報を得ることができないようにさせる．さらに，正規のアクセスポイントの代わりに同じ MAC アドレス，SSID でチャネルのみ異なる不正アクセスポイントを設置してビーコンを送信し，クライアントのプローブ要求に答える．その結果，クライアントが正規のアクセスポイントに接続しようとすると，意図せず攻撃者が設置した不正アクセスポイントに接続してしまうことになる．その後，攻撃者は妨害を停止して正規のアクセスポイントと接続し，双方からのパケットを転送することで中間者になることができる．

2.3 KRACKs

鍵再インストール攻撃（Key Reinstallation AttaCKs, KRACKs）は 2017 年に Vanhoef らによって提案された Wi-Fi で使用されるセキュリティプロトコルの鍵生成・共有の脆弱性を利用した攻撃である [1]．KRACKs の攻撃対象はアクセスポイントおよびクライアントが使用している鍵生成/共有プロトコルと暗号化プロトコルによっていくつかの種類に分けることができる．鍵生成/共有プロトコルは 4 ウェイハンドシェイク，グループ鍵ハンドシェイク，高速移行ハンドシェイクが対象となる．また暗号化プロトコルとして，CCMP の他に TKIP（Temporal Key Integrity Protocol），GCMP（Galois/Counter Mode Protocol）が挙げられる．本稿では前述の理由により 4 ウェイハンドシェイクを攻撃対象とした．また，暗号化プロトコルは WPA2 で標準採用されている CCMP を対象とした．

KRACKs は各ハンドシェイク内の特定のパケットを遮断したりタイミングをずらして送信したりすることで，暗号鍵の再インストールを引き起こし強制的にパケット番号のリセットを行わせる攻撃である．パケット番号がリセットされた結果，攻撃者は暗号化されたパケットの復号やリプレイ攻撃などが可能となる．攻撃者は特定のパケットの遮断や送信のタイミングをずらすためにチャネルベース中間者攻撃を行って中間者になる必要がある．

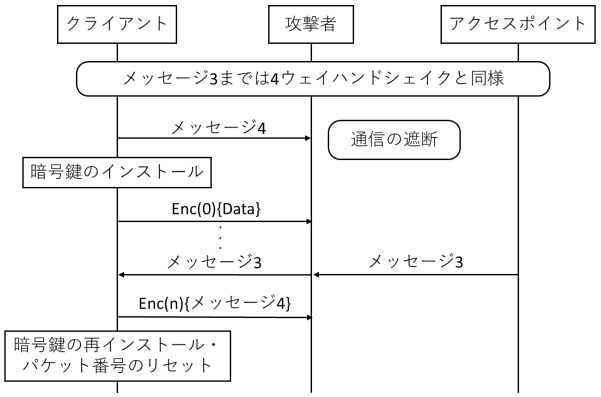

図 2 4 ウェイハンドシェイクに対する KRACKs

ここからは CCMP と 4 ウェイハンドシェイクに対する KRACKs の攻撃手法について説明する．4 ウェイハンドシェイクにおいて，アクセスポイントはメッセージ 3 を送信した後にメッセージ 4 を受け取れなかった場合，メッセージ 3 がロストしたと判断してクライアントにメッセージ 3 を再送する．クライアントは再送されたメッセージ 3 を受け取ると，メッセージ 4 を再送してからペア暗号鍵を再インストールする．この時，ペア暗号鍵が再インストールされるとパケット番号はリセットされる仕様のため，パケット番号は 0 に戻る．その結果，クライアントは元のペア暗号鍵と同じ鍵のまま，パケット番号を再利用してデータを暗号化してしまう．CCMP においてパケット番号の再利用は初期化ベクトルの再利用を意味する．つまり，2.1.2 節にあるように攻撃者は平文に関する情報を得ることができてしまう．ただし，クライアントがメッセージ 3 の再送によってペア暗号鍵を再インストールするかどうかは実装系によるといわれている．

実際の攻撃方法の一例を図 2 に示す．ただし，Enc(n) はパケット番号 n でペア暗号鍵を用いてデータが暗号化されていることを表す．攻撃者はチャネルベース中間者攻撃を行うことで中間者となり，メッセージ 4 のみを遮断する．クライアントはアクセスポイントから再送されるメッセージ 3 を受け取り，メッセージ 4 の再送とペア暗号鍵の再インストールおよびパケット番号のリセットを行う．

また，上記の他に攻撃者がメッセージ 3 を転送せずに保存しておき，アクセスポイントがメッセージ 4 を受け取れずに再送したメッセージ 3 と連続して送信することでペア暗号鍵を再インストールさせパケット番号をリセットさせる方法もある．

さらに，Ubuntu や Android など Linux 系の OS に対しては実装上のバグによってペア暗号鍵を再インストールする場合に元のペア暗号鍵ではなく全ての値が 0 の鍵をインストールする脆弱性（All-Zero Encryption Key Vulnerability）が発見されている [1]．この脆弱性がある場合は攻撃者はすべてのパケットを復号することができる．

表 1　実験に使用した機器

攻撃者	VMware 上で起動した Xubuntu Wi-Fi ドングル 3 台
アクセスポイント	モバイルルータ
クライアント	VMware 上で起動した Ubuntu 16.04 Windows 10 Android 6.0.1
Wi-Fi アダプタ	Wi-Fi アダプタ A（USB 接続） Wi-Fi アダプタ B（PC 内蔵）

3. チャネルベース中間者攻撃の評価実験

KRACKs の現実的な脅威を評価するためには，その前提となるチャネルベース中間者攻撃を評価する必要がある．本稿ではチャネルベース中間者攻撃の実現可能性について調べる二つの評価実験を行った．3.1 節では，様々な OS や Wi-Fi アダプタに対して，どのクライアントでなら攻撃が成功するのかを調査した．3.2 節では，それぞれの機器の位置関係など，どのような環境で攻撃が成功するのかを調査した．

3.1 中間者攻撃が実行可能なクライアント

3.1.1 実験方法

実験で使用した機器は表 1 の通りである．攻撃者はまず 3 つの Wi-Fi ドングルを接続する．それぞれアクセスポイントとの通信，クライアントとの通信，そしてアクセスポイントのチャネル妨害のために使用される．アクセスポイントはモバイルルータ，クライアントの OS は Ubuntu 16.04，Windows 10 および Android6.0.1 である．各 OS は KRACKs に対するパッチ適用後のものを用いた．Windows 10 に対しては Wi-Fi アダプタを 2 種類用意した．Ubuntu では VMware の仕様により内蔵の Wi-Fi アダプタは使用できず，USB 接続のものしか使用できなかったため，Wi-Fi アダプタ A のみを使用した．また，Android は本体内蔵の Wi-Fi アダプタを使用した．チャネルベース中間者攻撃を行うために Vanhoef らによって公開されているツール [7] を利用した．このツールは Xubuntu のコマンドライン上で実行可能であり，`channelmitm.cpp` を実行することで，中間者となって転送したパケットをファイルに保存し解析することが可能となる．

中間者攻撃が実行可能なクライアントを調べるために，各 OS および Wi-Fi アダプタの組み合わせに対して，攻撃者をアクセスポイントとクライアントの間に配置してチャネルベース中間者攻撃が成功するかを調べた．ただし，今回の評価実験の目的は WPA2 への KRACKs の影響を評価することである．そこで本実験においては攻撃者が 4 ウェイハンドシェイクのすべてのパケットを受信，転送に成功した場合を攻撃に成功したと定義した．

表 2　チャネルベース中間者攻撃が実行可能なクライアント

OS	Wi-Fi アダプタ	攻撃成功
Ubuntu16.04	Wi-Fi アダプタ A	○
Windows 10	Wi-Fi アダプタ A	○
Windows 10	Wi-Fi アダプタ B	×
Android 6.0.1	内蔵アダプタ	×

3.1.2 実験結果

実験の結果を表 2 に示す．Ubuntu と USB 接続の Wi-Fi アダプタ A に対してチャネルベース中間者攻撃を実行した結果として，クライアントが不正アクセスポイントに接続すると，すべての通信の履歴が攻撃者側で確認できたため，攻撃が成功していることが分かった．この時クライアントには，アクセスポイントが同じ SSID と MAC アドレスのままチャネルのみを変更したようにしか見えなかった．Windows 10 に USB 接続の Wi-Fi アダプタ A を接続した場合も同様の挙動が観測できたため攻撃に成功していることが分かった．どちらの場合も 4 ウェイハンドシェイクによってペア暗号鍵の生成/共有が行われた後も安定して通信することができた．

ノート PC 内蔵の Wi-Fi アダプタを使用した状態の Windows 10 では，チャネルベース中間者攻撃を行ってもクライアントは接続しようとしている正規のチャネルを表示し，正規のアクセスポイントに接続された．つまり，攻撃者は中間者になることができず攻撃に失敗した．中間者になることができなかったため転送したパケットの履歴もビーコンしか残っておらず，パケットの妨害なども不可能であった．

Android 6.0.1 に対してチャネルベース中間者攻撃を行った場合は正規のアクセスポイントのチャネルではなく攻撃者が設置した不正アクセスポイントのチャネルが表示されていた．しかし，アクセスポイントに接続しようとしたところ接続中の表示のままで，実際に接続することはできなかった．そこで Android に対して攻撃を行っている際のパケットを取得して調べた所，Android からプローブ要求が送信されており，攻撃者が設置した不正アクセスポイントがそれに対してプローブ応答を送信していたことが分かった．しかし，Android からそれに対する返答が送信されておらず不正アクセスポイントからのプローブ応答が破棄されていた．つまり，Android 機器に対してはチャネルベース中間者攻撃に失敗した．

3.1.3 考察

実験結果として，Windows 10 では内蔵の Wi-Fi アダプタと USB 接続のもので攻撃の成否が分かれた．つまり，チャネルベース中間者攻撃が実行可能かどうかはハードウェアにも依存する．従って，ハードウェアの面から KRACKs の対策が可能であるとも考えられる．

ただし，攻撃に対して脆弱性を持つようなクライアント

表 3　同室内でのチャネルベース中間者攻撃

パターン	成功確率	平均電波強度 [dBm]		
		クライアント		攻撃者
		正規 AP	不正 AP	
1	5/5	-63	-52	-58
2	5/5	-63	-56	-57
3	5/5	-59	-51	-64
4	5/5	-60	-51	-65

が中間者攻撃を検知するためには，アクセスポイントのチャネルが正規のものであるかを確認する必要があり，これは現実的には困難ではないかと考えられる．

3.2　中間者攻撃が実行可能な環境

チャネルベース中間者攻撃は攻撃者が物理的にクライアントおよびアクセスポイントの近くにいる必要があるとされる．そこで，攻撃者がどのような位置にいるときに攻撃が成功するのかを調べる実験を行った．

本稿ではアクセスポイントおよびクライアントと攻撃者の間の障害物の有無によって大きく 2 つのパターンに分けて実験を行った．3.2.2 節では障害物がなく，攻撃者がアクセスポイントおよびクライアントと同室にいる場合について攻撃が可能かを調査した．また，障害物がない場合に攻撃可能な距離を調査した．3.2.3 節では攻撃者が部屋の外にいる場合を想定して障害物がある環境で攻撃が可能かを調査した．

3.2.1　実験に使用する機器

実験に使用する機器は 3.1 節と同様である．ただし，クライアントは 3.1 節の結果より，チャネルベース中間者攻撃が成功した Windows 10 に USB 接続の Wi-Fi アダプタ A を接続したものを使用した．

3.2.2　同室内に攻撃者がいる場合

全員が同室内にいる場合での攻撃可能性について評価するために障害物のない 6 m × 3 m の部屋でチャネルベース中間者攻撃を行い，攻撃が成功するかを調べた．室内でのアクセスポイント，クライアント，攻撃者の配置は

1. アクセスポイント - 攻撃者 - クライアント
2. 攻撃者 - アクセスポイント - クライアント
3. 攻撃者 - クライアント - アクセスポイント
4. 等距離（正三角形に配置）

の 4 パターンを想定した．パターン 1 から 3 の配置では部屋の隅および中央に各機器を配置した．そして，それぞれの配置について 5 回ずつ実験を行った．各実験でクライアントは正規アクセスポイントと不正アクセスポイントの電波強度を，攻撃者は正規アクセスポイントの電波強度を記録し，最後に中間者攻撃が成功するかどうかを調べた．

その結果を表 3 に示す．ただし，AP はアクセスポイントを意味する．パターン 1 から 4 のどの配置であっても攻撃は 100%成功したため，同室内ではどのような配置であっても攻撃に成功することが分かった．またパターン 2 や 3 でも攻撃に成功したことから，攻撃者は物理的にアクセスポイントとクライアントの間に入る必要はなく，アクセスポイントとクライアントの距離が近くても攻撃に成功することが分かった．

表 4　障害物がある場合のチャネルベース中間者攻撃

パターン	成功確率	平均電波強度 [dBm]		
		クライアント		攻撃者
		正規 AP	不正 AP	
1	3/3	-65	-69	-79
2	3/3	-52	-60	-73
3	3/3	-49	-52	-75

次に，障害物がない環境ではアクセスポイントおよびクライアントと攻撃者がどの程度離れればチャネルベース中間者攻撃に失敗するかを調べるための実験を行った．

モバイルルータが実環境で使われるシーンを想定した場合，PC のすぐ近くに置くと考えられるため，アクセスポイントとクライアントの距離は 10 cm ほどに密接して配置した．そして攻撃者を徐々にアクセスポイントとクライアントから遠ざけつつチャネルベース中間者攻撃を行い，攻撃が失敗する距離を調べた．

その結果，攻撃者がアクセスポイントとクライアントから 40 m 離れたときにクライアントが取得するアクセスポイントのチャネルが正規のものと不正なものが切り替わりつつ表示されるようになり，接続しようとしてもすぐ通信が切れて安定しなくなった．つまり障害物がない場合は，攻撃者から 40 m ほど離れていなければ攻撃は成功してしまうということになる．

3.2.3　攻撃者が同室にいない場合

攻撃者がアクセスポイントおよびクライアントと同室内に存在せず，部屋の外からチャネルベース中間者攻撃を行う場合を想定して実験を行った．具体的には，攻撃者とアクセスポイントおよびクライアントの間に障害物がある環境で攻撃が成功するかを調べた．攻撃可能な距離を調べた時と同様にモバイルルータの使用を想定してアクセスポイントとクライアントを 10 cm ほどの距離にして障害物のない 6 m × 3 m の部屋に配置した．アクセスポイントおよびクライアントと攻撃者のあいだの障害物は 1. ガラス窓，2. ドア，3. コンクリート壁の 3 パターンとした．それぞれ，屋外，部屋の外，隣室からの攻撃を想定している．各モデルケースで攻撃を 3 回ずつ行い，同室内の実験と同様に電波強度を記録し，攻撃が成功するかどうかを調べた．

その結果を表 4 に示す．各機器の間に障害物がある時でもすべての場合について攻撃に成功した．また，正規アクセスポイントと不正アクセスポイントの電波強度を比較した結果，正規のアクセスポイントの電波強度が不正アク

セスポイントより強い場合でも攻撃に成功することが分かった．

3.2.4 考察

攻撃者が同室内にいる場合は必ず攻撃に成功し，通信も安定した．この場合は KRACKs のフェーズへ移ることも容易であると考えられる．クライアントとアクセスポイントを 10 cm ほどに近づけたとしても攻撃者が 40 m ほど離れるまでは攻撃に成功したため，会議室や商業施設の店舗などの大きな空間であっても障害物がない場合は攻撃が成功する確率は高いと懸念される．

また，攻撃者が窓や壁などの障害物を間に挟んだ場合でもチャネルベース中間者攻撃は成功した．つまり，攻撃者はアクセスポイントとクライアントがある部屋の隣室や外からでも中間者になることができる．

それぞれの環境でアクセスポイントの電波強度を測定した結果，たとえ正規のアクセスポイントの電波強度が不正アクセスポイントより強いとしても攻撃が成功する可能性があることも分かった．このような結果になったのは，攻撃者のジャミングによって正規のアクセスポイントのチャネルが妨害されているためであると考えられる．したがって，攻撃対象が据え置き型のルータなど，より電波強度が強い場合でも攻撃者のジャミングや偽のアクセスポイントの電波が届く場合は攻撃に成功する可能性が高い．

3.3 実環境での実現可能性と影響

チャネルベース中間者攻撃の評価実験を行った結果，攻撃に耐性のないクライアントについて，攻撃者の電波がクライアントに届く場合は容易に攻撃が実行可能であったため，チャネルベース中間者攻撃は実環境において十分実現可能であるといえることが分かった．被害者が中間者攻撃に気づくためには自分が接続しているアクセスポイントのチャネルを把握する必要がある．しかし一般的には，スマートフォンやノート PC でアクセスポイントを選択する際にチャネルは表示されない．従って，被害者がチャネルベース中間者攻撃を受けていることに気づくことは非常に難しい．また，攻撃者が中間者になることができればパケットの遮断・再送などが可能となるため，KRACKs に限らず，DoS 攻撃など他の攻撃も可能になるうえに，今後提案されるであろう新たな攻撃の脅威にさらされることになる．

以上の理由により，チャネルベース中間者攻撃は Wi-Fi に対する攻撃としては危険なものであり，KRACKs と同様に対策されるべき攻撃である．参考に Ubuntu16.04 や Windows10 は KRACKs に対してパッチによる対策がなされたが，3.1 節の結果から現在もチャネルベース中間者攻撃に対しては対策がされていないことがわかる．チャネルベース中間者攻撃を防ぐことができれば，同時に KRACKs に対する防衛策にもなり得る．そのため，今回の実験でチャネルベース中間者攻撃が失敗したクライアントについてさらに詳しく調査することで，対応策を見つける必要があると考える．

表 5　実験に使用した機器

攻撃者	VMware 上で起動した Xubuntu Wi-Fi ドングル 3 台
アクセスポイント	モバイルルータ
クライアント	VMware 上で起動した Ubuntu16.04 Windows10
Wi-Fi アダプタ	Wi-Fi アダプタ A

4. KRACKs の検証実験

本稿ではアクセスポイントとクライアントに対するチャネルベース中間者攻撃が必ず成功するという仮定の下で実環境を想定して KRACKs の検証実験を行った．具体的には，いくつかの OS に対して KRACKs による攻撃を行い，OS ごとの挙動を調べ，実際にパケットの復号が可能であるかどうかを調査した．

4.1 実験方法

実験で使用した機器は表 5 の通りである．3.1 節の結果からクライアントの OS は Ubuntu 16.04 と Windows 10 とし，どちらも Wi-Fi アダプタは USB 接続の Wi-Fi アダプタ A を使用した．なお，本実験では KRACKs が提案される前の環境を再現するためにパッチ適用後の Ubuntu16.04 と Windows10 に加えてパッチを適用していない Ubuntu16.04 も使用した．まず，攻撃者はチャネルベース中間者攻撃によってアクセスポイントとクライアントの間に割り込む．チャネルベース中間者攻撃を行う `channelmitm.cpp` [7] の中にある関数 `analyze_traffic`（受信したパケットの種類を判断する関数）と `handle_packet_ap`（パケットを転送するための関数）を改良することでクライアントに対して 2.3 節で述べた 2 通りの方法で KRACKs を検証した．その後，パケットの履歴を解析することでパケット番号がリセットされているかを確認した．さらに各 OS について，ペア暗号鍵の値が全て 0 の状態でパケットが暗号化されていないかの調査も行った．

4.2 実験結果

4.2.1 Ubuntu 16.04

VMware 上で起動したパッチによる対策がされる前の Ubuntu 16.04 に対して攻撃を行った．そしてパケット番号およびペア暗号鍵のリセットが発生するかどうかを調査した．表 6 がその結果である．どちらの攻撃方法でもペア暗号鍵のリセットが確認できたがパケット番号はリセットされなかった．

まず観測した遮断後のアクセスポイントとクライアント

表 6　KRACKs の検証実験の結果（Ubuntu 16.04 対策前）

	リセット対象	
	ペア暗号鍵	パケット番号
メッセージ 4 の遮断	○	×
メッセージ 3 の連続送信	○	×

表 7　KRACKs の検証実験の結果（Ubuntu 16.04 対策後）

	リセット対象	
	ペア暗号鍵	パケット番号
メッセージ 4 の遮断	×	×
メッセージ 3 の連続送信	×	×

の挙動を示す．メッセージ 3 までが攻撃者によって転送されたのち，メッセージ 4 がクライアントからアクセスポイントに向けて送信されたが攻撃者が転送しないためにアクセスポイントはメッセージ 4 を受信することができなかった．メッセージ 3 を受信したクライアントはペア暗号鍵をインストールした．その後，クライアントは通信が開始したと判断して DHCP リクエストなどのパケットを暗号化して送信し始めたが，アクセスポイントはペア暗号鍵をインストールしていないため復号することができずパケットをすべて破棄していた．メッセージ 3 の送信から一定時間が経過した後にアクセスポイントはメッセージ 3 がクライアントに届いていないと判断してメッセージ 3 を再送した．再送されたメッセージ 3 を受け取ったクライアントはペア暗号鍵で暗号化されたメッセージ 4 を送信した．その後，一度目のメッセージ 4 を送信した時と同じように通信を開始するためのいくつかのパケットを暗号化して送信し始めた．これらのパケットはペア暗号鍵ではなく，全ての値が 0 である暗号鍵で暗号化されていた．つまり，対策前の Ubuntu 16.04 にはペア暗号鍵が 0 でリセットされる脆弱性があることが確認できた．ここで再度送信されたメッセージ 4 は暗号化されていたため，先ほどまでと同様にアクセスポイントは通信を開始することができなかった．また，メッセージ 4 再送後のパケットのパケット番号はリセットされていなかった．

メッセージ 3 を連続して送信する方法でも攻撃を行った結果，メッセージ 4 を遮断した時と同様に再送されたメッセージ 4 はペア暗号鍵で暗号化されていた．その後の通信はすべての値が 0 の鍵で暗号化されており，パケット番号はリセットされていなかった．

また，パッチによる対策がされた後の Ubuntu16.04 に対しても同様の実験を行った．表 7 がその結果である．送受信されたパケットは対策前と同じであったが，ペア暗号鍵とパケット番号はリセットされておらず，パケットを復号することはできなかった．

4.2.2　Windows 10

Windows 10 を搭載したノートパソコンに対して Ubuntu 16.04 と同様の攻撃を行い，その挙動を調べた．その実験の結果を表 8 に示す．どちらの攻撃方法でもペア暗号鍵およびパケット番号のリセットは確認できなかった．

表 8　KRACKs の検証実験の結果（Windows 10）

	リセット対象	
	ペア暗号鍵	パケット番号
メッセージ 4 の遮断	×	×
メッセージ 3 の連続送信	×	×

4 ウェイハンドシェイクのメッセージ 4 を遮断してメッセージ 3 を再送させたところ，クライアントはメッセージ 4 の再送を行わなかった．その後クライアントから送信されたパケットはすべて元のペア暗号鍵によって暗号化されており，パケット番号のリセットもされていなかった．また，アクセスポイントはメッセージ 4 を受信できないためペア暗号鍵をインストールすることができず，通信を開始することはできなかった．

メッセージ 3 を保存して再送する攻撃でも，先ほどと同様にクライアントはメッセージ 4 を一度しか送信せず，そのまま通信を開始するためのパケットを送信し始めた．それらのパケットは元のペア暗号鍵で暗号化されており，パケット番号もリセットされていなかった．この場合はアクセスポイントはメッセージ 4 を受け取ることができるため通信が開始された．つまり，Windows 10 ではどちらの方法で攻撃しても通信が開始しない以外の影響はなく，KRACKs によってパケットを復号することはできなかった．

4.3　Vanhoef らの結果との相違

KRACKs の検証実験を行った結果，Vanhoef らの論文 [1] で述べられた結果と異なる点を二つ発見した．

一つ目は鍵を再インストールしてもパケット番号がリセットされなかった点である．Vanhoef らは鍵が再インストールされた際には必ずパケット番号がリセットされると述べている．パケット番号がリセットされることでパケットの復号が可能になるため，パケット番号がリセットされなければパケットの復号ができず KRACKs による影響はないと考えられる．今回の実験では対策前の Ubuntu 16.04 に対して攻撃を行った際，鍵を再インストールしてもパケット番号がリセットされていないことを確認した．Ubuntu にはペア暗号鍵の値がすべて 0 になるという脆弱性があるため，パケットの復号は可能であったが，その脆弱性を持たない OS でも同様の結果となった場合，パケット番号がリセットされなかったという事実はすべてのパケットの復号が不可能になることを意味する．

二つ目は特定の条件下で KRACKs を行うと，通信が継続できなくなる点である．4.2.1 節で述べたようにメッセージ 4 を遮断する方法で KRACKs を行うと，再送されたメッ

セージ 4 はペア暗号鍵で暗号化されていた．一方，アクセスポイントはペア暗号鍵をインストールしていなかったためクライアントからのパケットを受け取れずアクセスポイントは通信を開始しなかった．この問題についてはメッセージ 3 を連続して送信する方法によって解決することができる．しかし，クライアントにペア暗号鍵がリセットされる脆弱性がある場合は仮にメッセージ 4 をアクセスポイントが受信してペア暗号鍵をインストールしたとしても，クライアントとアクセスポイントで暗号鍵が異なるために通信することができない．つまり，KRACKs によって暗号鍵の値をすべて 0 にリセットすることができたとしてもクライアントは DHCP リクエストなどの通信を開始するためのパケットしか送信せず通信が続かないため攻撃者はクライアントから情報を得ることはできない．

4.4 実環境での実現可能性と影響

実験の結果，今回想定したモデルケースでは KRACKs によってパケットが復号可能となり，暗号化された通信内容が解析されてしまう状況は非常に限られており，一部の OS に対してしか有効とは言えないことが分かった．しかし，Ubuntu 16.04 に対しては暗号鍵がすべて 0 にリセットされる脆弱性が存在し，容易にパケットを復号できることも分かった．既存の攻撃手法では 4 ウェイハンドシェイクが完了せず，DHCP リクエストなど一部のパケットを送信するだけで通信が途切れてしまったため，不正アクセスポイントで通信を続けさせるなど攻撃の改良が必要である．従って，提案された手法をそのまま実行してパケットの復号を行い続けるのは実環境では不可能に近く，攻撃を改良していくことで一部の OS に対しては通信内容を傍受することが可能になる．ただし，現在ではパッチによる対策がされており，正しくアップデートをしておけば通信内容を傍受される心配はない．

KRACKs への対策として，Windows 10 のようにメッセージ 3 の再送を受け付けないようにさせたり，ペア暗号鍵を再インストールした際にパケット番号をリセットさせないことが挙げられる．現状パケットの復号は一部の OS 以外では難易度が高く，それらの OS でも今回発見したような問題点が存在するため 4 ウェイハンドシェイクに対しての KRACKs による影響は少ないと考えられる．しかし旧バージョンでは通信を傍受される可能性があるため，ユーザーが確実にアップデートなどを行うことで KRACKs の対策をすることが望まれる．

5. まとめと今後の課題

本稿では，Wi-Fi の利用環境を想定し，WPA2 のセキュリティプロトコルである AES-CCMP の 4 ウェイハンドシェイクに対する KRACKs の現実的な脅威度を評価した．評価の方法として実際に攻撃を行い，攻撃が成功するかどうかを調査した．また，攻撃の過程で必要となるチャネルベース中間者攻撃についてもその脅威度を調べた．

実験の結果，攻撃者がクライアントの付近に来ることができる場合はチャネルベース中間者攻撃を行ってクライアントに気づかせずに中間者になるのは容易であることが分かった．また，実環境を想定して 4 ウェイハンドシェイクに対する KRACKs を行ったところ，一部 OS でパケットの復号に成功したのみであり現実的には成功する環境が限られていること，復号に成功する場合でも攻撃の改良が必要であることが分かった．また，現在ではパッチによる対策がされているため適切なアップデートを行えば攻撃者はパケットを復号することができない．

今回は会議室の中などクライアントが移動せずアクセスポイントとユニキャスト通信をする場合を想定した．今後は，社内を移動するケースを想定し，アクセスポイントを乗り換えるローミングに対する検討を行うために，高速移行ハンドシェイクに対する KRACKs の評価と対策を行っていきたい．また，Wi-Fi において脅威になると考えられるチャネルベース中間者攻撃への対応策を見つけるために，攻撃が失敗したクライアントについてさらに詳しく調査していきたい．

参考文献

[1] Vanhoef, M. and Piessens, F.: Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2, *ACM SIGSAC CCS 2017* (Thuraisingham, B. M., Evans, D., Malkin, T. and Xu, D., eds.), ACM, pp. 1313–1328 (2017).

[2] Vanhoef, M. and Piessens, F.: Advanced Wi-Fi attacks using commodity hardware, *ACSAC 2014* (Jr., C. N. P., Hahn, A., Butler, K. R. B. and Sherr, M., eds.), ACM, pp. 256–265 (2014).

[3] IEEE-SA: 802.11i-2004 - Amendment 6: Medium Access Control (MAC) Security Enhancements.

[4] IEEE-SA: 802.11-2016 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[5] IEEE-SA: 802.1X-2010 - Local and metropolitan area networks - Port-Based Network Access Control.

[6] IEEE-SA: 802.11r-2008 - Amendment 2: Fast Basic Service Set (BSS) Transition.

[7] URL: `https://github.com/vanhoefm/modwifi`.