情報処理学会論文誌 Vol.53 No.9 2225–2233 (Sep. 2012)

推薦論文

Bluetoothのセキュアシンプルペアリングに対する中間者攻撃

野村 大翼[1] 松尾 和人[1,2]

受付日 2011年11月29日, 採録日 2012年3月2日

概要：情報端末等の接続に用いられる無線通信方式である Bluetooth では，端末間で相互認証を行う手順をペアリングと呼ぶ．Bluetooth 2.1 + EDR において規定されたセキュアシンプルペアリングは，盗聴と中間者攻撃に対する安全性が仕様に謳われているペアリング方式である．セキュアシンプルペアリングのモードの 1 つである Passkey Entry モードは，端末に入力されたパスキーの一致によって相互認証を実現する．最近，Lindel と Phan-Mingard によって，Passkey Entry モードに対する中間者攻撃が提案された．しかし，これらの攻撃法では，攻撃者がペアリングを複数回アボートし，そのたびにユーザにペアリングを再試行させる必要があった．また，再試行の際に，ユーザが毎回同一のパスキーを入力することを仮定した，実現性の低い攻撃法であった．本論文では，Bluetooth のセキュアシンプルペアリングの Passkey Entry モードに対する新たな中間者攻撃を提案する．本論文で提案する攻撃法は，Lindel，Phan-Mingard の攻撃法の改良手法であり，ペアリング再試行時に以前と異なるパスキーの入力を許す，より現実的な攻撃法である．本論文では，提案攻撃法への対策についても考察する．

キーワード：Bluetooth，ペアリング，セキュアシンプルペアリング，Passkey Entry モード，中間者攻撃

A Man-in-the-Middle Attack against Secure Simple Pairing in Bluetooth

Daisuke Nomura[1] Kazuto Matsuo[1,2]

Received: November 29, 2011, Accepted: March 2, 2012

Abstract: This paper proposes a man-in-the-middle attack against the passkey entry mode in the secure simple pairing for Bluetooth. Bluetooth is a wireless communication system used to connect between digital devices. The mutual authentication procedure between Bluetooth devices is called pairing. The secure simple pairing is a series of the pairing specified in Bluetooth 2.1 + EDR. The passkey entry mode achieves the authentication by entering the same passkey into the devices. Its specification claims the passkey entry mode is secure against eavesdropping and man-in-the-middle attacks. Recently, Lindel, Phan and Mingard proposed man-in-the-middle attacks against this mode. These attacks need to abort the pairing procedure and to make the user to retry it. Moreover, the attacks require that the user enters the same passkey as before in the retries. The proposed attack allows the different passkey in the retry, so that it is more practical than the previous attacks. This paper also discusses the countermeasures against the proposed attack.

Keywords: Bluetooth, pairings, secure simple pairing, Passkey Entry mode, man-in-the-middle attacks

1. はじめに

Bluetooth [1], [2], [3], [4] は，情報端末間または情報端末

[1] 情報セキュリティ大学院大学情報セキュリティ研究科
Graduate School of Information Security, Institute of Information Security, Yokohama, Kanagawa 221–0835, Japan
[2] 神奈川大学理学部情報科学科
Department of Information Sciences, Faculty of Science, Kanagawa University, Hiratsuka, Kanagawa 259–1293, Japan

本論文の内容は 2011 年 10 月のコンピュータセキュリティシンポジウム 2011（CSS2011）にて報告され，同プログラム委員長により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である．

とキーボード等の周辺機器を接続する際に用いられる無線通信方式であり，多くの携帯端末やゲーム機等に採用されている．Bluetooth の規格は，まず 2004 年に Bluetooth 2.0 + EDR が登場し，2007 年に Bluetooth 2.1 + EDR が策定された．2009 年には Bluetooth 3.0 + HS，2010 年には Bluetooth 4.0 が策定されたが，現在は 2.0 + EDR と 2.1 + EDR が普及している．

Bluetooth 端末どうしを通信可能な状態にするために，端末間で相互認証し，関連付けを行うことをペアリングと呼ぶ．ペアリングは 2.0 + EDR までは PIN を用いた方式のみであったが，2.1 + EDR においてセキュアシンプルペアリング（以下では SSP と呼ぶ）が追加された．

PIN によるペアリングでは，実際は 4 桁の固定された PIN が多用されていたうえ，PIN が機器の Web 上で閲覧可能な取扱い説明書に書かれていることが多く，攻撃者が PIN を容易に知り得た．したがって，PIN によるペアリングの安全性は Bluetooth 登場直後から懸念されていた．2001 年に Jakobsson ら [5] によって，PIN によるペアリングの脆弱性が指摘され，攻撃法が提案された．2005 年には Shaked ら [6] によって，Jakobsson らの攻撃法が実装された．実際に Shaked らは，4 桁の PIN を Pentium III 450 MHz を搭載した PC を用いて 0.27 秒で導出した．

PIN によるペアリングの脆弱性に対応したものが SSP である．SSP には Numeric Comparison，Just Works，Out of Band，Passkey Entry という 4 種類のモードがある．Passkey Entry モードは，k bit 整数値をとるパスキーを端末に入力するモードであり，ユーザからは PIN によるペアリングと同じに見える．Passkey Entry モードは盗聴と中間者攻撃に対して安全であることが仕様書 [3, Vol.1, Part A, pp.57–58] に謳われている．だが近年，SSP に対する攻撃法も登場している．まず，形式検証ツール ProVerif [7], [8] を用いた安全性検証で，Numeric Comparison モードに対する攻撃 [9] が確認されている．また，Just Works モードを利用する攻撃法 [10], [11] と，Lindell や Phan らによって提案された Passkey Entry モードに対する攻撃法 [12], [13] が知られている．後者の Passkey Entry モードに対する攻撃法は，端末に入力されたパスキーが 1 bit ごとに k 回検証される仕組みを攻撃に利用している．しかし，この攻撃法は攻撃者がペアリングを複数回アボートしてユーザにペアリングを再試行させる必要があり，ペアリング再試行の際にユーザが毎回同一のパスキーを入力するという仮定を必要とするため，実現性の低い攻撃法であった．

本論文では，Passkey Entry モードに対する新たな中間者攻撃を提案する．提案する攻撃法は，前述した Lindel, Phan らが提案した既知の攻撃法の改良手法であり，攻撃者が端末間の通信を中継・改竄し，両端末になりすまして他方の端末とのペアリングを完了する攻撃法である．既知の攻撃法は攻撃者がペアリングを複数回アボートする必要があるが，提案攻撃法は必要アボート回数が 1 回である．加えて，ペアリング再試行の際にユーザが以前と同一のパスキーを入力する必要がなく，異なるパスキーの入力を許す．したがって，既知の攻撃法と比較してより現実的な攻撃法であると考えられる．

本論文ではまず，2 章で SSP を紹介し，3 章で SSP の Passkey Entry モードに対する既知の攻撃法を紹介する．そして 4 章で新たな中間者攻撃法を提案し，5 章ではその攻撃法への対策を考察する．最後に 6 章で本論文をまとめる．

2. セキュアシンプルペアリング

Bluetooth を利用する端末どうしを相互認証し，関連付ける手順がペアリングである．ペアリングによって，認証や暗号鍵生成に用いられるリンクキーと呼ばれる重要情報が生成され，端末間で共有される．

本章では，ペアリングの一方式である SSP の概略を説明する．

SSP は 5 つのフェーズからなる．フェーズ 1 では楕円曲線 Diffie-Hellman 鍵共有（ECDH）によって公開鍵を共有し，フェーズ 2 ではフェーズ 3，4 で用いられる値を共有する．フェーズ 3 では共有したすべての値が端末間で正しく共有できていることを検証する．フェーズ 4 でリンクキーを生成し，フェーズ 5 で認証・暗号化を行う．**図 1** に，SSP の流れを示す．

以下では [3, Vol.2, Part H] に従って端末 A，B のペアリングのフェーズ 1–3 の概略を説明する．各フェーズで用いられるハッシュ関数等の詳細についても [3, Vol.2, Part H] を参照されたい．

2.1 フェーズ 1：公開鍵交換

フェーズ 1 では公開鍵の交換を行い，端末間で Diffie-Hellman 鍵 $DHKey$ を共有する．フェーズ 1 の流れを**図 2** に示す．

まず，端末 A と B は ECDH の秘密鍵・公開鍵のペア (SK_a, PK_a)，(SK_b, PK_b) を各々生成し，公開鍵を相手の端末に送信する．端末 A，B は自身の秘密鍵と相手から受信した公開鍵を鍵生成関数 P192 に入力し，$DHKey$ の生成を行う．

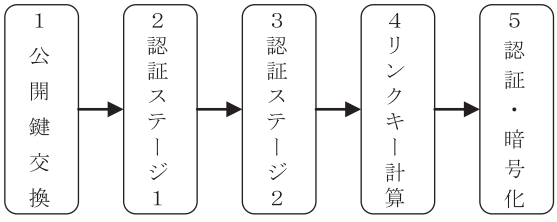

図 1 SSP の流れ
Fig. 1 Sequence of Secure Simple Pairing.

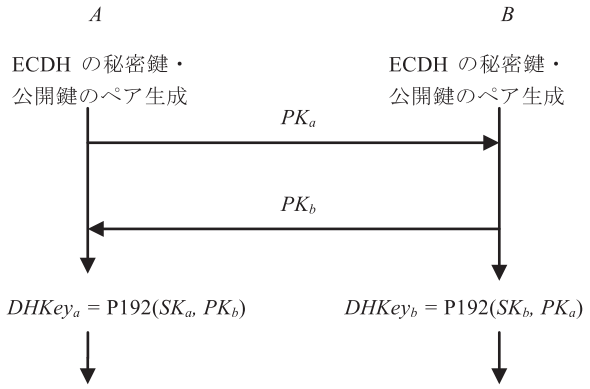

図 2 公開鍵交換
Fig. 2 Sequence of public key exchange.

A: $DHKey_a = \mathrm{P192}(SK_a, PK_b)$

B: $DHKey_b = \mathrm{P192}(SK_b, PK_a)$

鍵生成関数 P192 の仕様については [3, Vol.2, Part H, pp.897–898] を参照されたい．

フェーズ 1 を終了すると端末 A，B で $DHKey$ が共有される．すなわち，

$$DHKey_a = DHKey_b$$

となる．

DH 鍵交換プロトコルは中間者攻撃に対する耐性がないが，この後のフェーズで中間者攻撃への対応が行われる．

2.2 フェーズ 2：認証ステージ 1

フェーズ 2 には前述の Numeric Comparison，Just Works，Out of Band，Passkey Entry の 4 つのモードが規定されており，端末が持つユーザインタフェース I/O によって使用モードが選択される．以下では，Passkey Entry モードについて詳細に説明する．

Passkey Entry モードは，以下に示す 2 種類のモードに細分される [3, Vol.2, Part F, p.707].

モード 1：
端末 A，B がともにディスプレイを持たず，キーボードのみを持つ場合に対応する．ユーザが同一のパスキーを端末 A，B に入力する．

モード 2：
端末 A がディスプレイのみを持ち，端末 B がキーボードのみを持つ場合に対応する．端末 A がランダムに生成・表示したパスキーをユーザが端末 B に入力する．

なお，**モード 1** が選択される場合は端末 A，B がともにキーボードであることが想定されるが，著者らの知る限り**モード 1** が実装された例はない．

以下では，**図 3** に従って Passkey Entry モードの手順を説明する．

A，B に表示・入力された k bit のパスキーを各々

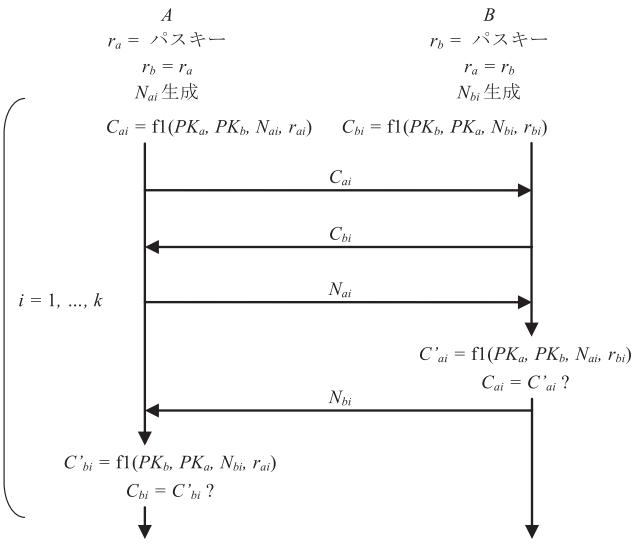

図 3 Passkey Entry モードの手順
Fig. 3 Procedure of Passkey Entry mode.

$$r_a = (r_{a1}, r_{a2}, \ldots, r_{ak}),\ r_{ai} \in \{0,1\}$$

$$r_b = (r_{b1}, r_{b2}, \ldots, r_{bk}),\ r_{bi} \in \{0,1\}$$

とする．

Passkey Entry モードでは，パスキーが一致していることを，ビットごとに検証する．すなわち $r_{ai} = r_{bi}$ を $i = 1, \ldots, k$ に対して検証する．以下に，i bit 目の検証手順を示す．

A はナンス N_{ai} を生成し，PK_a，PK_b，r_{ai} と N_{ai} に対しハッシュ関数 f1 を用いてハッシュ値

$$C_{ai} = \mathrm{f1}(PK_a, PK_b, N_{ai}, r_{ai})$$

を計算し，B に送信する．ハッシュ関数 f1 の仕様については [3, Vol.2, Part H, pp.898–899] を参照されたい．B はナンス N_{bi} を生成し，PK_a，PK_b，r_{bi} と N_{bi} に対するハッシュ値

$$C_{bi} = \mathrm{f1}(PK_b, PK_a, N_{bi}, r_{bi})$$

を計算し，A に送信する．

A は N_{ai} を B に送信する．N_{ai} を受信した B は PK_a，PK_b，r_{bi}，N_{ai} を用いて A と同じ計算を行い，

$$C'_{ai} = \mathrm{f1}(PK_a, PK_b, N_{ai}, r_{bi})$$

を得る．次に B は $C_{ai} = C'_{ai}$ を検証する．$C_{ai} \neq C'_{ai}$ ならば $r_{ai} \neq r_{bi}$ であるとして，B はペアリングをアボートし，$C_{ai} = C'_{ai}$ であれば $r_{ai} = r_{bi}$ であるとして，N_{bi} を A に送信する．A は PK_a，PK_b，r_{ai}，N_{bi} を用いて B と同じ計算を行い，

$$C'_{bi} = \mathrm{f1}(PK_b, PK_a, N_{bi}, r_{ai})$$

を得る．A は $C_{bi} = C'_{bi}$ を検証する．$C_{bi} \neq C'_{bi}$ ならば $r_{ai} \neq r_{bi}$ であるとして，A はペアリングをアボートし，

$C_{bi} = C'_{bi}$ であれば $r_{ai} = r_{bi}$ であるとして，i をインクリメントする．

以上を $i = k$ まで繰り返し，r_a，r_b の各ビットが同一であることを検証する．たとえば，パスキーが 6 桁入力された場合は $k = 20$ であり，20 回の繰返しが必要となる．検証が k 回行われた後に，N_{ak}，N_{bk} を各々 N_a，N_b として，フェーズ 3 に渡す．

2.3 フェーズ 3：認証ステージ 2

このフェーズでは，フェーズ 1，2 で交換した値が端末間で正しく共有されていることを検証する．フェーズ 1，2 から引き継がれた $DHKey_a$，$DHKey_b$，N_a，N_b，r_a，r_b と，端末 A，B の I/O 情報 $IOcapA$，$IOcapB$，Bluetooth MAC アドレス BD_ADDR_a，BD_ADDR_b を用いる．端末の I/O 情報と Bluetooth MAC アドレスはフェーズ 1 以前に交換済みである．

A はハッシュ関数 f3 によってハッシュ値

$$E_a = \text{f3}(DHKey_a, N_a, N_b, r_b, IOcapA, BD\_ADDR_a, BD\_ADDR_b)$$

を計算し，B に送信する．ハッシュ関数 f3 の仕様については [3, Vol.2, Part H, pp.900–901] を参照されたい．B は A から E_a を受信し，A と同じ計算を行って

$$E'_a = \text{f3}(DHKey_b, N_a, N_b, r_b, IOcapA, BD\_ADDR_a, BD\_ADDR_b)$$

を得る．$E_a \neq E'_a$ ならば共有した値が異なっているとして，B はペアリングをアボートする．$E_a = E'_a$ ならば，B はハッシュ関数 f3 によってハッシュ値

$$E_b = \text{f3}(DHKey_b, N_b, N_a, r_a, IOcapB, BD\_ADDR_b, BD\_ADDR_a)$$

を計算し，A に送信する．

A は B から E_b を受信し，

$$E'_b = \text{f3}(DHKey_a, N_b, N_a, r_a, IOcapB, BD\_ADDR_b, BD\_ADDR_a)$$

を計算し，$E_b = E'_b$ を検証する．$E_b \neq E'_b$ ならば共有した値が異なっているとして，A はペアリングをアボートする．$E_b = E'_b$ ならば，フェーズ 4 へ進む．

フェーズ 4，5 については [3, Vol.2, Part H, p.897]，[3, Vol.2, Part H, pp.868–880] を参照されたい．

フェーズ 5 を終了後，端末 A，B は通常の通信を開始する．また，リンクキーをすでに共有している場合は，フェーズ 5 から認証手順が開始される．詳細については [3, Vol.2, Part F, pp.691–698] を参照されたい．

3. 既知の攻撃法

本章では，Passkey Entry モードに対する既知の攻撃法を紹介する．Passkey Entry モードに対する攻撃法は Lindell [12] や Phan ら [13] により提案された．既知の攻撃は以下の **A1**，**A2** に分類される．

A1：
ペアリングの手順を盗聴し，盗聴で得られたデータを用いてパスキーを計算・入手する攻撃

A2：
攻撃者がターゲットの端末のペアリング時に正規の端末を装ってペアリングを試み，パスキーを計算・入手する中間者攻撃

これらの攻撃法はともにユーザにペアリングを複数回再試行させる必要があり，その際にユーザが毎回同一のパスキーを入力する必要がある．また，この仮定を満足するためには，ユーザがパスキーを選択する必要があり，必然的に（実装例のない）**モード 1** に対してのみ有効な攻撃となる．

両攻撃はともに Passkey Entry モードにおいて，C_{ai}，C_{bi} を計算する際にパスキー r_a，r_b が r_{ai}，r_{bi} として 1 bit ごとに検証される仕組みを利用した攻撃である．

前者の攻撃は，攻撃者が端末 A，B のペアリング過程を盗聴し，盗聴で得られたデータからパスキーを計算・入手する攻撃である．盗聴した PK_a，PK_b と N_{ai}，0 をハッシュ関数 f1 に入力し，

$$C''_{ai} = \text{f1}(PK_a, PK_b, N_{ai}, 0)$$

を計算する．$C_{ai} = C''_{ai}$ なら $r_{ai} = 0$ であり，$C_{ai} \neq C''_{ai}$ なら $r_{ai} = 1$ であることが分かる．この $C_{ai} = C''_{ai}$ の検証を $i = 1, \ldots, k$ に対して繰り返し行い，得られた r_{ai} $(i = 1, \ldots, k)$ を連結すればパスキー r_a $(= r_b)$ が得られる．パスキーを得た攻撃者は，ペアリングを再試行する際にユーザが以前と同一のパスキーを利用した場合には，得られた r_a を用いて端末 A，B とのペアリングが可能であるため，中間者攻撃につながる．

後者の攻撃は，ターゲットの端末のペアリング時に正規の端末になりすましたうえで A–B の通信を中継・改竄し，それぞれに Passkey Entry モードでペアリングを試みることによって，パスキーを計算・入手する中間者攻撃である．攻撃者 M は B になりすまして A とのペアリングを実行し，A になりすまして B とのペアリングを実行する．以下では **A2** の攻撃の概略を説明する．

まずフェーズ 1 において，A，B，M は ECDH の秘密鍵・公開鍵のペア (SK_a, PK_a)，(SK_b, PK_b)，(SK_m, PK_m) を各々生成する．次に A は PK_a を M に送信する．M は A から PK_a を受信し，PK_m を PK_b として A に送信する．A と M は以下を計算し，M–A で $DHKey$ を共有する．

A: $DHKey_a = \mathrm{P192}(SK_a, PK_m)$

M: $DHKey_{ma} = \mathrm{P192}(SK_m, PK_a)$

また，M は B に PK_m を PK_a として送信し，B は PK_a（$= PK_m$）を受信した後，M に PK_b を送信する．B と M は以下を計算し，M–B で $DHKey$ を共有する．

B: $DHKey_b = \mathrm{P192}(SK_b, PK_m)$

M: $DHKey_{mb} = \mathrm{P192}(SK_m, PK_b)$

次に，フェーズ 2 において，A は

$$C_{ai} = \mathrm{f1}(PK_a, PK_m, N_{ai}, r_{ai})$$

を計算し，M に送信する．M は C_{ai} を受信し，ナンス N_{mi} を生成した後に

$$C_{mai} = \mathrm{f1}(PK_m, PK_b, N_{mi}, 0)$$

を計算し，C_{ai} として B に送信する．B は

$$C_{bi} = \mathrm{f1}(PK_b, PK_m, N_{bi}, r_{bi})$$

を計算し，M に送信する．M は

$$C_{mbi} = \mathrm{f1}(PK_m, PK_a, N_{mi}, 0)$$

を計算して C_{bi} として A に送信する．A は N_{ai} を M に送信する．N_{ai} を受信した M は N_{mi} を N_{ai} として B に送信する．

B は

$$C'_{ai} = \mathrm{f1}(PK_m, PK_b, N_{mi}, r_{bi})$$

を計算し，受信した $C_{ai} = C'_{ai}$ を検証する．B は，もし $C_{ai} = C'_{ai}$ ならば N_{bi} を M に送信し，$C_{ai} \neq C'_{ai}$ ならばペアリングをアボートする．

B が N_{bi} を送信した場合は $r_{bi} = 0$ であり，B がペアリングをアボートした場合は $r_{bi} = 1$ であることが分かる．M はこの値を記録する．

N_{bi} を受信した M は N_{mi} を N_{bi} として A に送信する．A は

$$C'_{bi} = \mathrm{f1}(PK_m, PK_a, N_{mi}, r_{ai})$$

を計算し，受信した $C_{bi} = C'_{bi}$ を検証する．$C_{bi} = C'_{bi}$ ならば検証を通過し，$C_{bi} \neq C'_{bi}$ ならば A はペアリングをアボートする．実際には，$r_{ai} = r_{bi}$ であるため，B がペアリングをアボートしない限り，A もアボートしない．

上記の M–A と M–B のフェーズ 2 を $i = 1, \ldots, k$ まで繰り返す．A，B にペアリングをアボートされた場合でも，ユーザがペアリングを再試行する際に同じ攻撃を行い，その際にユーザが以前と同一のパスキーを入力すれば，記録されたパスキーのビット値を利用してさらにパスキーを蓄積していくことができる．$i = 1, \ldots, k$ までの繰返しが終わった後にビット値を連結すればパスキー r_b（$= r_a$）が得られる．パスキーを得た攻撃者は，ターゲットの端末が再びペアリングを行う際にユーザが以前と同一のパスキーを入力すれば，A，B とのペアリングを成功させることが可能になる．

この攻撃では平均して $k/2$ 回のペアリングの試行回数でパスキーの全 bit が得られる．

4. 提案攻撃法

本章では，前章で説明した攻撃 **A2** を改良し，Passkey Entry モードに対する新たな中間者攻撃を提案する．

提案攻撃法は，**A2** と同様に M が B になりすまして A とペアリングを実行し，A になりすまして B とペアリングを実行する．その際に，本来の A–B 間の通信とは同期をずらしたうえで，A から受信した値を B から送信された値として A に送り返す等の操作を行い，r_{ai} を導出するとともに，A，B の検証式を成立させる．そして M–B のペアリングを成功させるとともに，M–A のペアリングをアボートする．その際，ユーザは B が M とペアリング済みであることを気付かずに M–A のペアリングを再試行する可能性が高い．そこで，M–B の通信を維持した状態でユーザにペアリングを再試行させ，M–A のペアリングを成功させる．そして，M–A，M–B のペアリングを完成させる．

4.1 攻撃の手順

ここでは，6 ステップに分けて提案攻撃法の手順を示す．ステップ 1 は M–A，M–B のフェーズ 1 に対応した手順である．次のステップ 2，3 は $i = 1, \ldots, k$ に対して繰り返される．まず，ステップ 2 で M–A のフェーズ 2 のパスキー 1 bit ごとの検証から r_{ai} を導出し，ステップ 3 ではステップ 2 で導出した r_{ai} を用いて M–B でのフェーズ 2 を行う．そしてステップ 4 において M–A のペアリングをアボートし，ステップ 5 で M–B のペアリングを成功させる．最後のステップ 6 では M–A のペアリングを再試行の後に成功させ，中間者攻撃を完成させる．**図 4** に，提案攻撃法のステップ 1–3 の概略を示す．図 4 において，M から A，B への矢印は，矢印の終点の値に対して，実際には矢印の始点の値を M が送信することを意味する．すなわち，A，B は始点の値を終点の値として用いてペアリングを実行する．以下に攻撃手順の詳細を示す．

(1) ステップ 1：公開鍵交換

攻撃者 M は ECDH の秘密鍵·公開鍵のペア (SK_m, PK_m) を生成する．M は A から PK_a を受信し，PK_a として PK_m を B に送信する．B は M に PK_b を送信し，

$$DHKey_b = \mathrm{P192}(SK_b, PK_m)$$

を生成する．M は B から PK_b を受信し，

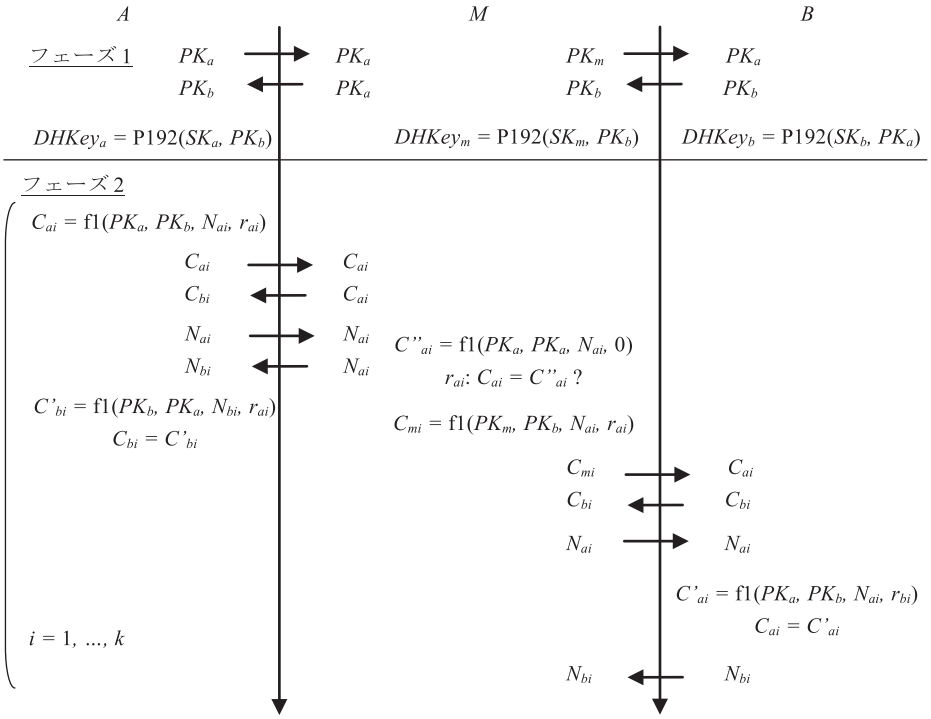

図 4 提案攻撃法（ステップ 1–3）
Fig. 4 Proposed attack (Step 1–3).

$$DHKey_m = \mathrm{P192}(SK_m, PK_b)$$

を生成する．ここで，

$$DHKey_m = DHKey_b$$

となる．次に，M は A から受信した PK_a を PK_b として A に送り返す．したがって，A が生成する $DHKey_a$ は実際には以下のようになる．

$$DHKey_a = \mathrm{P192}(SK_a, PK_a)$$

その後，A はパスキー r_a をランダムに生成しユーザは r_a を B に r_b として入力する，あるいはユーザは A と B に同一のパスキー r_a，r_b を入力する．

(2) ステップ 2：r_{ai} の導出

A から C_{ai} を受信した後に M は C_{ai} を C_{bi} として A に送り返す．すると，A は N_{ai} を M に送信する．A から N_{ai} を受信した後に，M は PK_a，N_{ai}，0 をハッシュ関数 f1 に入力して，

$$C''_{ai} = \mathrm{f1}(PK_a, PK_a, N_{ai}, 0)$$

を計算する．そして，M は $C_{ai} = C''_{ai}$ を検証することで r_{ai} を導出する．すなわち，$C_{ai} = C''_{ai}$ ならば $r_{ai} = 0$ であり，$C_{ai} \neq C''_{ai}$ ならば $r_{ai} = 1$ であるとする．この r_{ai} の値をステップ 3 に引き継ぐ．

次に M は N_{ai} を N_{bi} として A に送り返す．A は C'_{bi} を計算し，$C_{bi} = C'_{bi}$ を検証する．A が受信した値はそれぞれ $C_{bi} \leftarrow C_{ai}$，$PK_b \leftarrow PK_a$，$N_{bi} \leftarrow N_{ai}$ であり，実際には

$$C'_{bi} = \mathrm{f1}(PK_a, PK_a, N_{ai}, r_{ai})$$

である．したがって，A には

$$C_{bi} = \mathrm{f1}(PK_a, PK_a, N_{ai}, r_{ai}) = C'_{bi}$$

と見えるため，A の検証を通過する．

(3) ステップ 3：r_{ai} を M–B のフェーズ 2 で利用

M は，PK_m，PK_b および A から受信した N_{ai} とステップ 2 で導出した r_{ai} を使って

$$C_{mi} = \mathrm{f1}(PK_m, PK_b, N_{ai}, r_{ai})$$

を計算して，C_{ai} として B に送信する．B は

$$C_{bi} = \mathrm{f1}(PK_b, PK_m, N_{bi}, r_{bi})$$

を計算し，M に送信する．

B から C_{bi} を受信した後に，M は A から受信した N_{ai} を B に送信する．

B は

$$C'_{ai} = \mathrm{f1}(PK_m, PK_b, N_{ai}, r_{bi})$$

を計算し，B は $C_{ai} = C'_{ai}$ を検証する．

M はステップ 2 で得た正しい r_{ai} を用いて

$$C_{mi} = \mathrm{f1}(PK_m, PK_b, N_{ai}, r_{ai}) \ (\rightarrow C_{ai})$$

を計算しているため必ず $C_{ai} = C'_{ai}$ となり，この検証は通過する．最後に B は M に N_{bi} を送信する．

M は $i = 1, \ldots, k$ まで，ステップ 2，3 をアボートせずに

表 1 既知の攻撃法と提案攻撃法の比較
Table 1 Comparison of proposed attack and known attacks.

	既知の攻撃 [12][13]	提案攻撃法
仮定	ペアリングを再試行する際に，ユーザが以前と同一のパスキーを入力する	端末 B が攻撃者とペアリングを完了し通信を継続していることにユーザが気付かない
攻撃対象	実装例なし(*KB-KB*)	実装例あり(*PC-KB*, etc.)
アボート回数	約 $k/2$ 回	1 回

繰り返すことで，M–A，M–B のフェーズ 2 を通過する．

(4) ステップ 4：A とのペアリングアボート

M–A のペアリングはフェーズ 3 で失敗する．なぜならば，M は A の PK_a に対応する秘密鍵 SK_a を知らないため，フェーズ 1 で A との $DHKey$ の共有ができず，フェーズ 3 で M が B を装って A に送信すべき値 E_b を計算することができないからである．そこで，M は適当な値を E_b として A に送信し，A にペアリングをアボートさせる．

(5) ステップ 5：B とのペアリング成功

M は

$$E_m = \mathrm{f3}(DHKey_m, N_a, N_b, r_b, IOcapM, BD\_ADDR_m, BD\_ADDR_b)$$

を計算し，E_a として B に送信する．ここで，$IOcapM$ は M の I/O 情報，BD_ADDR_m は Bluetooth MAC アドレスである．B は E'_a を計算し，$E_a = E'_a$ を検証する．B が受信した値はそれぞれ $E_a \leftarrow E_m$，$IOcapA \leftarrow IOcapM$，$BD\_ADDR_a \leftarrow BD\_ADDR_m$ であるので，実際には

$$E'_a = \mathrm{f3}(DHKey_b, N_a, N_b, r_b, IOcapM, BD\_ADDR_m, BD\_ADDR_b)$$

であり，さらに $DHKey_m = DHKey_b$ である．したがって，B には $E_a = E'_a$ と見え，この検証を通過する．

B の検証は以降も通過し，M–B 間のペアリングは成功する．ペアリング成功後も M は B との通信を維持する．

(6) ステップ 6：A とのペアリングの再試行

A がペアリングをアボートしたため，ユーザは A–B 間のペアリングが失敗したと考え，ペアリングを再試行する．その際，ユーザは B が通信を継続中であることに気付かないものとする．Passkey Entry モードが選択された場合，B はディスプレイを持たないため，この仮定は十分な妥当性を有すると考えられる．ユーザは，A が新たに生成したパスキー r'_a を r'_b として B に入力する，あるいは新たなパスキー r'_a，r'_b を A と B に入力する．ここで，$r'_a = r'_b$ である．このとき，B と M は通信中であるため，M は B に入力された r'_b を B から入手できる．したがって，M は B から入手した r'_b を用いて Passkey Entry モードの正当な手順を踏むことで A とのペアリングにも成功し，中間者攻撃に成功する．

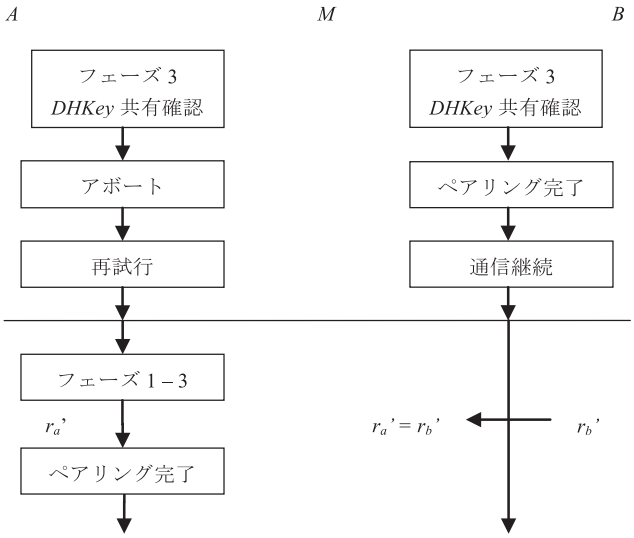

図 5 提案攻撃法（ステップ 4–6）
Fig. 5 Proposed attack (Step 4–6).

提案攻撃法のステップ 4–6 の概略を図 5 に示す．

4.2 既知の攻撃法と提案攻撃法の比較

既知の攻撃法 **A2** と提案攻撃法の比較を表 1 に示す．

提案攻撃法は，既知の攻撃法と同様に，ペアリングを攻撃者がアボートしてユーザにペアリングを再試行させる必要があるが，再試行のたびにユーザが以前と同一のパスキーを入力するという，既知の攻撃が必要とする仮定を必要とせずに攻撃を行うことが可能である．一方，提案攻撃法は，端末 A がペアリングをアボートした後に，端末 B が攻撃者と通信を継続していることにユーザが気付かない必要がある．この仮定は，既知の攻撃法に必要な上記仮定と比較して，より現実的な仮定であると考えられる．なぜならば，端末 B はディスプレイを持たないキーボードであり，ユーザインタフェースに乏しいためである．B が通信状態を表示する LED を備えている場合等には，ユーザが通信継続中であることに気付き，攻撃シナリオに沿った行動をとらないことも考えられる．しかし，その場合にも M は A になりすまして B とのペアリングに成功しており，

攻撃の脅威は依然として残っていると考えられる．

また，既知の攻撃は，実装例のない（両端末がともにキーボードである）**モード 1** のみが攻撃対象であった．一方，提案攻撃法は，既知の攻撃法が必要とする仮定を必要としないので，PC とキーボードの組合せ等の実装例のある**モード 2** に対しても有効な攻撃である．

さらに，既知の攻撃はパスキーの全ビット数の約半分 $k/2$ 回のペアリングのアボート回数が必要であるが，提案攻撃法では 1 回であり，この点からも実現性の高い攻撃手法であると考えられる．

5. 提案攻撃法への対策

本章では，4 章で提案した中間者攻撃への対策を考察する．

Phan ら [13] は既知の攻撃法 [12], [13] への対策として，フェーズ 2 においてパスキーを 1 bit ごとに検証するのではなく全 k bit を一括して検証することをあげている．この対策によって平均して 2^{k-1} 回のアボート–再試行が必要となり，既知の攻撃に対してこの対策は有効である．提案攻撃法も既知の攻撃法と同様にパスキーを 1 bit ごとに検証する仕組みを利用している．したがって，この対策は提案攻撃法に対しても有効であると考えられる．ただし，既知の攻撃では端末からの応答を観察してパスキー推定を行っていたが，提案攻撃法では攻撃者の内部計算によってパスキー推定を行っているので，この対策を施すことで攻撃者の応答時間は増大するものの，アボート–再試行回数は 1 回のままである．したがって，対策を施した後にも，パスキーのビット長 k を，提案攻撃法が存在しない場合と比較してより長くとる必要がある．また，Passkey Entry モードは 1 bit ごとに k 回に分けてパスキーを検証することで，可変長パスキーに対応しているが，全ビットの検証を一括して行うためにはハッシュ関数 f1 の拡張等の仕様の大幅な変更が必要となる．

既知の攻撃法と違い，提案攻撃法では攻撃者 M と端末 A の 1 度目のペアリングにおいて M が PK_a を PK_b として A に送り返している．したがって，このときに A が $PK_b \neq PK_a$ を検証し，$PK_b = PK_a$ である場合にペアリングをアボートすれば，提案攻撃法は成立しない．ECDH のランダムな公開鍵が同一になる確率はきわめて低く，この検証は実用的な影響を与えない．また，この対策は Phan らの対策と比較して小規模な仕様変更で実現可能である．したがって，この対策は提案攻撃法への最も効果的な対策であると考えられる．

6. まとめ

本論文では Bluetooth のセキュアシンプルペアリングの Passkey Entry モードに対する新たな中間者攻撃を提案した．提案攻撃法は，ペアリングを攻撃者がアボートしてユーザにペアリングを再試行させる必要があるが，再試行の際にユーザが以前と同一のパスキーを入力するという既知の攻撃が必要とする仮定を必要とせず，以前と異なるパスキーの入力を許すより現実的な攻撃である．また，既知の攻撃ではペアリングを複数回アボート–再試行する必要があったが提案攻撃法では 1 回であり，さらに既知の攻撃法は実装例が存在しないモードに対するものであったが，提案攻撃法は実装例が存在するモードに対しても適用可能である．本論文では，提案攻撃法への対策についても考察した．

今後の課題として提案攻撃法の実装があげられる．提案攻撃法や既知の中間者攻撃 [12], [13] は，端末間の正規の通信を遮断する必要がある．しかし，Bluetooth は短距離無線通信の規格であるため，端末間通信を遮断可能な状況は限られる．したがって，攻撃の現実的な脅威を確認するためには，提案攻撃法の実装実験が必要である．また，パスキーの一括検証による対策において，所望の安全性を得るために必要なパスキーのビット長の指針が，提案攻撃法の実装実験から得られると考えられる．

謝辞 本論文に関して有益なご助言をいただいた匿名査読者各位に感謝いたします．

参考文献

[1] Bluetooth SIG: Bluetooth 4.0 Core Specification, Bluetooth SIG (2009), available from ⟨https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737⟩ (accessed 2011-11-11).

[2] Bluetooth SIG: Bluetooth 3.0 + HS Core Specification, Bluetooth SIG (2009), available from ⟨https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=174214⟩ (accessed 2011-11-11).

[3] Bluetooth SIG: Bluetooth 2.1 + EDR Core Specification, Bluetooth SIG (2007), available from ⟨https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=241363⟩ (accessed 2011-11-11).

[4] Bluetooth SIG: Bluetooth 2.0 + EDR Core Specification, Bluetooth SIG (2004), available from ⟨https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=40560⟩ (accessed 2011-11-11).

[5] Jakobsson, M. and Wetzel, S.: Security weakness in Bluetooth, *Topics in Cryptology – CT-RSA2001*, LNCS 2020, pp.176–191, Springer (2001).

[6] Shaked, Y. and Wool, A.: Cracking the Bluetooth PIN, *Proc. 3rd USENIX/ACM Conf. Mobile Systems, Applications, and Services* (*MobiSys2005*), pp.39–50 (2005).

[7] Blanchet, B. et al.: ProVerif: Cryptographic protocol verifier in the formal model (online), available from ⟨http://www.proverif.ens.fr/⟩ (accessed 2011-11-11).

[8] Blanchet, B., Abadi, M. and Fournet, C.: Automated verification of selected equivalences for security protocols, *J. Logic and Algebraic Programming*, Vol.75, No.1, pp.3–51 (2008).

[9] Chang, R. and Shmatikov, V.: Formal Analysis of Authentication in Bluetooth Device Pairing, *Proc. Joint Workshop on Foundations of Computer Security and Automated Reasoning for Security Protocol Analysis*

(*FCS-ARSPA2007*), pp.45–61 (2007).
[10] Haataja, K. and Hypponen, K.: Man-In-The-Middle attacks on Bluetooth: A comparative analysis, a novel attack and countermeasure, *Proc. 2008 3rd International Symposium on Communications, Control and Signal Processing* (*ISCCSP 2008*), pp.1096–1102 (2008).
[11] Haataja, K. and Toivanen, P.: Two Practical Man-In-The-Middle Attacks on Bluetooth Secure Simple Pairing and Countermeasure, *IEEE Trans. Wireless Communications*, Vol.9, No.1, pp.384–392 (2010).
[12] Lindell, Y.A.: Attacks on the Pairing Protocol of Bluetooth v2.1, Blackhat USA 2008 (2008), available from ⟨http://www.blackhat.com/presentations/bh-usa-08/Lindell/BH_US_08_Lindell_Bluetooth_2.1_New_Vulnerabilities.pdf⟩ (accessed 2011-11-11).
[13] Phan, R. and Mingard, P.: Analyzing the Secure Simple Pairing in Bluetooth v4.0, *Wireless Personal Communication*, DOI: 10.1007/s11277-010-0215-1, Springer (2010), available from ⟨http://www.springerlink.com/content/730t872p4t720180/⟩ (accessed 2011-11-11).

推薦文

本論文は，Bluetooth のセキュアシンプルペアリングの Passkey Entry モードに対する新たな中間者攻撃を提案している．近年，Lindel や Phan，Mingard によって同モードに対する中間者攻撃法が提案されているが，彼らの攻撃法では，攻撃者がペアリングを複数回アボートしてユーザにペアリングを再試行させる必要があり，その際にユーザが毎回同一のパスキーを入力することを仮定していた．本論文では，彼らの中間者攻撃法を改良して，必要アボート回数が 1 回であり，ユーザは必ずしも以前と同じパスキーを入力する必要はない，というさらに現実的な環境下での中間者攻撃法を提案している．また，本論文では，提案攻撃法に対する対策も考察されている．本論文は，Bluetooth のセキュアシンプルペアリングに対する安全性の向上に，新たな攻撃法およびその対策の両面から貢献すると考え，推薦論文として推薦する．

（コンピュータセキュリティシンポジウム 2011
プログラム委員長　四方順司）

野村 大翼

2010 年麗澤大学国際経済学部国際経済学科卒業．2012 年情報セキュリティ大学院大学情報セキュリティ研究科博士前期課程修了．

松尾 和人　（正会員）

1988 年中央大学大学院理工学研究科博士前期課程電気工学専攻修了．同年東洋通信機（株）入社．2001 年中央大学大学院理工学研究科博士後期課程情報工学専攻修了．博士（工学）．2002 年中央大学研究開発機構機構助教授．2003 年同機構教授．2004 年情報セキュリティ大学院大学教授．2012 年より神奈川大学理学部情報科学科教授，情報セキュリティ大学院大学客員教授．電子情報通信学会，応用数理学会，国際暗号学会各会員．