Link Key取得について



Bluetoothログ解析方法

Bluetooth機器の通信は暗号化されており、解析時には暗号化を 解く "Link Key"が必要となります。Sodera又はBPA600を使用した ログ解析方法は以下の2通りです。

1. 測定機器のどちらかをSSP Debug Modeに設定する

2. Android端末の「btsnoop_hci.log」ファイルから Link Key を取得する



【方法1】 測定機器をSSP Debug Modeに設定する

Bluetooth機器は SSP Debug Mode と呼ばれる開発モードに設定可能です。 Master/Slave機器のいずれかをSSPデバッグモードに設定すると、Link Key を入力せずに 暗号化以降の解析が可能になります。

SSP Debug Mode とは

● Link Keyは通常、コネクション毎に毎回異なるKeyが生成されますが、 Bluetooth機器を SSP Debug Mode に設定することにより、Link Key 構成要素の一部が固定になります。

そのため、CPASソフトウェアは Link Keyを自動認識 (計算)することが 可能になります。

● <u>SSP Debug Modeへの切替方法は各セットメーカーにより異なり、</u> <u>また対応していないBluetooth機器もあるため、必ず各セットメーカーへ</u> ご確認ください。



【方法1】 SSP Debug Modeに設定した場合の表示例

Master/SlaveのいずれかがSSP Debug Mode になっていると、Link keyを入力しなくても 暗号化以降のタブが表示されます。

● 333 3 3 encapsulated_header accepted Slave master 11 ● 396 3 encapsulated_payload accepted Slave master 12 ● 407 3 encapsulated_payload accepted Slave master 11 ● 407 3 encapsulated_payload accepted Slave master 11 ● 410 3 encapsulated_payload * Master * master 16 * 3 encapsulated_payload * Master * master 11 ● accepted Slave master 11 10 0 11 10 0 Pichtred_rate Slave master 11 10 0 11 0 0 10 0 10 0		D	Frame#	LT_Addr	Original Opcode	Opcode	Role	Initiated by	Fram
● 396 3 encapsulated_payload Master master 26 ● 407 3 encapsulated_payload accepted Slave master 11 ● 410 3 encapsulated_payload master master 26 ● 410 3 encapsulated_payload Master master 26 ● 410 3 encapsulated_payload Master master 26 ● 410 3 encapsulated_payload Master master 11 ● 410 3 encapsulated_payload * Master * master 26 ● 410 3 encapsulated_payload * Master * master 26 ● accepted Slave master 11 11 10 11 ● 11 1 0 1 1 0 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 1 0		•	393	3	encapsulated_header	accepted	Slave	master	11
● 407 3 encapsulated_payload accepted Slave master 11 ● 410 3 encapsulated_payload Master master 26 ● 410 3 encapsulated_payload accepted Slave master 11 *3 encapsulated_payload accepted Slave master 11 *3 encapsulated_payload *Master *master 26 accepted Slave master 11 *3 encapsulated_payload *Master *master 26 accepted Slave master 11 11 0 11 11 10 10 11 10 10 0 11 10 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 <t< td=""><td></td><td>•</td><td>396</td><td>3</td><td></td><td>encapsulated_payload</td><td>Master</td><td>master</td><td>26</td></t<>		•	396	3		encapsulated_payload	Master	master	26
410 3 encapsulated_payload Master master 26 Slave master 11 * Master * master 26 Slave master 11 * Master * master 26 Slave master 11 * Master * master 11 * P.192 Public Key Public Key		•	407	3	encapsulated_payload	accepted	Slave	master	11
115 3 encapsulated_payload accepted Slave master 11 *3 encapsulated_payload *Master *master 26 accepted Slave master 11 preferred_rate Slave master 11 preferred_rate Slave master 13 18: (Master) Len=26 that the data were reconstructed. N 0 0 0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0		•	410	3		encapsulated_payload	Master	master	26
*3 encapsulated_payload ** Master ** master 26 Slave master 11 preferred_rate Slave slave 11 encapsulated_header Slave master 13 18: (Master) Len=26 that the data were reconstructed. id: Role: Master *Address: 3 *Opcode: LMP_encapsulated_payload ** Transaction ID: Initiated by master 9 of b 5 e 4 7 9 9 f			**5	3	encapsulated_payload	accepted	Slave	master	11
k key (Debug Key) は ncapsulated payload ny h内で確認できます。				* 3		encapsulated_payload	* Master	* master	26
k key (Debug Key) は acapsulated payload" かいト内で確認できます。 * Address: 3 -* Opcode: LMP_encapsulated_payload -* Transaction ID: Initiated by master -* P.192 Public Key					payload	accepted	Slave	master	11
3 encapsulated_header Slave master 13 incapsulated payload" 18: (Master) Len=26 8 0 1 1 1 1 1 0 0 1 1 1 0 1 0 1 1 0 1 0 1 0 1 0 0 1 1 0 1 0 1 0 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 0 0 1	k kov (Dobug Ka					preferred_rate	Slave	slave	11
ICapSulated payload 18: (Master) Len=26 why 内で確認できます。 18: (Master) Len=26 that the data were reconstructed. 00010001100011000 d: 000100010000 R 7 c ed f 8 25 11 R 7 c ed f 8 25 11 Proprode: LMP_encapsulated_payload -* Transaction ID: Initiated by master P.192 Public Kein	k key (Debug ke	=y) (&		3		encapsulated_header	Slave	master	13
・ ・ that the data were reconstructed. NOOOOIOOIIOOOIIIOOOIIIOOOIIIOOOIIIOOOIIIOOOIIIOOOIIIOOOIIIOOOIIIOOOIIIO Role: Master - *Address: 3 - * 15 25 11 -*Opcode: LMP_encapsulated_payload - * 9 15 2 6 5 8 -*Transaction ID: Initiated by master - * 0 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 0 10 0 10 10 10 10 0 0 10 <td></td> <td>Uau</td> <td>18: (Mz</td> <td>aster) Len=26</td> <td></td> <td>B 0</td> <td>11111</td> <td>00 11</td> <td>101</td>		Uau	18: (Mz	aster) Len=26		B 0	11111	00 11	101
Image: dt: Image: Constraint of the second seco	ット内で確認できま	9.	that th	ne data were re	constructed.	NO	00100	01 101	100
Role: Master F 7 c e d f 8 2 5 1 1 *Address: 3 *Cpcode: LMP_encapsulated_payload D f e c 3 9 f 6 f 5 8 *Transaction ID: Initiated by master P 9 d b 5 e 4 7 9 9 f *P-192 Public Key P *P P *P *P *P			nd:			1	10100	10 001	101
Role: Master Dfec39f6f58 *Address: 3 X1525eaf7b9 *Opcode: LMP_encapsulated_payload P9db5e4799f *Transaction ID: Initiated by master Pb0 D:*P-192 Public Key Pb0						R 7	ed	f 8 2 5	1 1
*Address: 3*Opcode: LMP_encapsulated_payload*Transaction ID: Initiated by master*P-192 Public Keu		-	Role: Ma	ster		Ôf	e c 3	9 f 6 f	5 8
* Opcode: LMP_encapsulated_payload * Transaction ID: Initiated by master * P_192 Public Key			*Address:	3		× 1	5 2 5	ea f7	b 9
Transaction ID: Initiated by master			* Opcode:	LMP_encapsul	ated_payload	. 9	d b 5	e4 79	9 f
E-*P.192 Public Key			* Transacti	on ID: Initiated	by master	An	n		1000
			* D 100 D.	blic Keu		N ~	-		
		6	Debug	Keu(X): 0x 15	20 70 09 98 44 21 a6 58 6	9f c3 fe 7e 43 29 d2 8			

【方法2】 Android端末から Link Keyを抜き出す

Android端末との通信を解析する場合、Android端末内に記録される 「btsnoop_hci.log」ファイルから、Link Keyを抜き出すことが可能です。

抜き出したLink KeyをCPASソフトウェアに入力することで 暗号化以降のプロトコル解析も可能になります。

※本手法はAndroid端末(OS ver 4.4以降)のみ有効であり、 <u>iPhone端末は非対応</u>です。



btsnoop_hci.log 取得方法 (1)

<ログ取得前の確認事項>

- CPASソフトウェアが最新バージョンであることを確認する
- Android端末のペアリング履歴・デバイス情報を完全削除する
- 一度端末をPCに接続し、古い「btsnoop_hci.log」が残っていないかを確認する
 ⇒残っている場合は、削除する





btsnoop_hci.log 取得方法 (2)

 ・端末の「設定」画面を開く
 →最下層の「端zz末情報」
 →「ビルド番号」を7回タップ

🖿 0 🖵 📾 📽 📽 🖛 💷 👘 🚮 🚳 17:29 < · 端末情報 201511月1月1日 使用情報 匿名のデータ/使用統計情報をSony Mobileに送 信します モデル番号 SO-01G ブロセッサー情報 Qualcomm MSM8974PRO-AC Androidバージョン 4.4.4 ベースバンドバージョン 8974-AAAAANAZQ-00009-06 カーネルバージョン 3.4.0 Built あなたは今開発者になりました! ビルド番号 23.0.B.1.38 ŵ • D

②①で開発者モードになり 「設定」画面に開発者向けオプションが 表示される



 3 開発者向けオプションをONに変更し、 Bluetooth HCIスヌープログを 「有効にする」をチェック

💊 開発者向け	オブション		0
パグレポートも	を取得		
PCバックアッ PCフルバックアッ は削除する場合に	ブバスワー ブ用のバスワ 選択します	ドードを変更	また
スリーブモート	ドにしない		
Bluetooth HCI/パケ Bluetooth HCI/パケ ルにキャプチャす	スヌープログを ットをすべて る	有効に ファイ	
プロセスの統計 プロセスの実行に	片情報 関する独自の	A211174 PR	
デバッグ			
USB 世 い SB 接続時 は デバッグ	ッグモードに	する	
USBデバッグの	の許可の取	り消し	
1935 d = ~ ~ (N/SZEL-E	- 14	
•⊃	ଜ	٥	
	Tech	nnolo	ogi
			-

btsnoop_hci.log 取得方法 (3)

- ④ 端末とペアリングを行う
- 5 端末を再起動
- ⑥ 再度端末をPCに接続し、「btsnoop_hci.log」をPCに保存





取得した btsnoop_hci.log から Link Keyを確認

① CPASソフトウェアを起動し、「Open Capture File...」を選択



② 拡張子を「BT Snoop Files.log」に変更し、PCに保存した 「btsnoop_hci.log」ファイルを開く





抜き出したLink Keyの表示例 (1)

成功例

"Link key"を検索すると、以下の通り表示されます。

Frame Display - btsnoop_hci.log

File Edit View Format Filter Bookmarks Options Wind	ow Help					
🗞 🊰 🔎 🗄 🔩 🔻 🐺 😂 🎜 🏥 📗	🛄 📖 🗾	🛃 🔟 📀				
Frame 942: (Controller) Len=26 		₿ © 	0	Find: link	key 🗸	🖗 🔎 🖉 Summary: HC
- Packet from: Controller	HLI UANT HLI	LZLAP SUP	RFLU	MM AVUIP		Media Hands-Free A2DF
- HCI Event	B Frame#	HCI Packet Type	Fram	Delta	Timestamp	
Event: Link Key Notification	942	Event	26	00:00:00.3	2017/05/18 7:55:45.401328	
Total Length: 23	943	Event	6	00:00:00.0	2017/05/18 7:55:45.403720	
- Bluetooth Device Address: 0x8c-de-52-24-3d-8b	944	ACL Data	17	00:00:00.0	2017/05/18 7:55:45.442433	
	945	ACL Data	21	00:00:00.0	2017/05/18 7:55:45.538386	
MAR. 0x32	946	ACL Data	21	00:00:00.0	2017/05/18 7:55:45.540721	
Link Key 0xd3 7e 8b 5b 60 8t be 34 te 61 be t7 do 23 84 43	947	ACL Data	21	00:00:00.0	2017/05/18 7:55:45.605164	
LINK NEV TYDES, ONBUMERICALED COMPINATION NEV	948	Event	8	00:00:00.0	2017/05/18 7:55:45.605945	
	949	ACL Data	19	00:00:00.0	2017/05/18 7:55:45.607329	
	950	ACL Data	19	00:00:00.0	2017/05/18 7:55:45.607784	
	951	ACL Data	29	00:00:00.0	2017/05/18 7:55:45.609120	
	952	Event	8	00:00:00.0	2017/05/18 7:55:45.674843	
	953	ACL Data	22	00:00:00.0	2017/05/18 7:55:45.742211	
	954	ACL Data	119	00:00:00.3	2017/05/18 7:55:46.078505	
	955	ACL Data	17	00:00:00.0	2017/05/18 7:55:46.080891	

※検証機器: Android OS 6.0.1のXperiaとスマートスピーカーとの接続



抜き出したLink Keyの表示例 (2)

失敗例 エラー表示となり、Link Keyが表示されない

Frame Display - btsnoop_hci.log							
File Edit View Format Filter Bookmarks Options W	/indow	Help					
🚷 🚰 🔎 표 😼 🝸 🏹 😂 🏖 🟥 🛛	I		🛾 💿 🔟 📈				
Frame 933: (Controller) Len=9			BOOC	າຍ	Find: link	key 🗸 💭 💭 💭	
HCI - Bluetooth Device Address: Field Truncated or Not Present	llofi						13
HCI UART:	HCI	UART HC	Data				
HCI Packet Type: Event Packet		-				-	_
E-HU:	B	Frame#	HCI Packet Type	Fram	Delta	Timestamp	
Packet from: Controller		913	Event	7	00:00:00.0	2018/01/05 5:10:25.735459	
		914	Command	8	00:00:00.0	2018/01/05 5:10:25.735706	
Tetal Lanche C		915	Event	9	00:00:00.0	2018/01/05 5:10:25.736239	
Physical Poly in Address: Field Typested or Mat Present		916	Command	8	00:00:00.0	2018/01/05 5:10:25.736527	
Bidetooth Device Address. Fleid Trancated of Not Flesent		917	Event	9	00:00:00.0	2018/01/05 5:10:25.737061	
		918	Event	6	00:00:00.0	2018/01/05 5:10:25.737293	
		919	ACL Data	21	00:00:00.0	2018/01/05 5:10:25.738366	
		920	Event	8	00:00:00.0	2018/01/05 5:10:25.744679	
		001	C 1	~	00.00.00.0	0010 101 105 5 40 05 345000	

※以下の場合はセットメーカー様へお問合せ下さい。

- 複数回試してもエラー表示が続いた場合
- ●「btsnoop_hci.log」ファイル自体が取得できない場合



ご不明点などございましたら、お気軽にご連絡ください

お問合せ先

〒105-0014 東京都港区芝3丁目5番1号コーンズハウス コーンズテクノロジー株式会社 電子通信ソリューション営業部 TEL:03-5427-7566 Email:ctl-comm@cornes.jp

