智慧型手機之 SQLite 資料庫刪除之特徵研究: Line 應用案例說明
Research of Smartphone's SQLite database deleted pattern:Case study on Line App)

林書宇、黃敬博、鄭宇軒
鑒真數位有限公司
albert@iforensics.com.tw

摘要

本研究主要探討主流通訊軟體(Line、WeChat、Whatsapp、Skype)之 SQLite 資料庫之資料庫特徵,針對遭到刪除但並未抹除的資料庫之 Hex 特徵進行研究。

關鍵詞：SQLite DB、DB Hex 特徵(Data base hex pattern)、資料庫(database)、Line、APP 鑑識(APP forensics)、手機鑑識(mobilephone forensics)。

壹、 前言

隨著近年來各類型行動裝置(智慧型手機、平板電腦等)的蓬勃發展，幾乎人人都擁有個人的智慧型行動裝置。而人與人的訊息交流也逐漸轉變到如今使用各種 APP 通訊軟體的習慣。

而通訊軟體的便利性也成為許多犯罪者的聯絡工具,因此許多案件的偵查經常會需要去檢視通訊軟體的聊天紀錄以找尋線索或是犯罪的證據。而在講究個人資料保護及隱私權的現代社會中,通訊軟體通常都能夠刪除聊天紀錄以保護個人隱私，但這樣的功能卻增添了不少辦案的困難。

行動裝置的種類繁多，而各種手機數據的儲存方式也多少有所差異，一般大眾主流使用的 Android(HTC、Samsung、小米等等)、iOS(Apple iPhone)作業系統通常將使用之數據紀錄儲存在 database 中。本研究主要對於 SQLite 資料庫的 Hex 特徵進行研究及分析，以期能夠回復行動裝置刪除之數據，為行動鑑識取證分析的研究提供實務上蒐證的參考。

貳、 文獻探討

如前段所提到的，現今智慧型行動裝置的便利也造就了一股從行動裝置中衍生而出的各種通訊軟體的熱潮，而一般主流通訊軟體 Line、Whatsapp、WeChat、Skype 等皆是使用 SQLite 資料庫儲存訊息。因此若要研究如何回復刪除的數據，必須先明白到底 SQLite 資料庫是什麼。

2.1 SQLite 資料庫

SQLite 資料庫顧名思義是一種精簡版、簡化版的 SQL 語法的一種資料庫，由於其輕巧的特性，通常會使用於嵌入式的裝置(例如智慧型手機等等)。因為 SQLite 擁有著占用很少資源、操作方便以及資料庫儲存結構簡單的優勢，因此對於智慧型手機這種硬體設備受限的裝置以及這類型裝置上的應用程式 App 大部分都使用 SQLite 作為資料儲存的方式。

2.2 SQLite 資料儲存結構

SQLite 資料庫的儲存結構大致上是使用 B-tree 的儲存結構，亦即資料的儲存通常會建立在一個類似樹狀的結構，其中包含了 Root Page、Internal Page、Leaf Page(請詳見下圖 1)。Root Page 就是樹的根部，其內容儲存了關於檔案大小以及連向 Internal Page 的指標，通常只有一個 Root Page。Internal Page 可以理解為類似樹枝的定位，其裡面也是沒有儲存資料的，只有存了 key 以及連向 Leaf Page 的指標(Pointer)。而 Leaf Page 則是本研究所要研究的重點所在，因為其為所有資訊儲存的位置。

圖 1：SQLite 資料庫樹狀架構圖

2.3 Leaf Page

Leaf Page 其中包含了已刪除資料所存在的未分配區域(unallocate area)和夾在部分刪除但未整頁刪除的 freeblock 區域。其中 freeblock 區域是一個很特別的區域,在 SQLite 資料庫的架構中,若是一個 page 部分資料被刪除而不是整個 page 被刪除，這些刪除的資料會夾在一些未刪除的資料之間，但這整個 page 大小不變,page 也不會因此消失,而這夾在未刪除資料之間的區塊就是 freeblock 區塊。簡而言之,若要找到已刪除但未遭到覆蓋的資料,則必須先找到所對應之樹根、樹枝,最後再前往定位的葉子頁去找尋 freeblock 區塊和 unallocate 區塊進行操作以復原資料。

參、針對刪除仍未被抹除的資料庫之 Hex 特徵研究

3.1 SQLite 資料庫結構

主流通訊軟體例如”Line”、”Wechat”、”Skype”或”WhatsApp”等均為使用 SQLite Database 來儲存訊息之手機通訊軟體,詳細資訊如表 1。而本研究以”Line”為例，實際操作回復 SQLite Database 中被刪除的訊息，使用之工具為 Micro Systemation XACT。

表 1：常用通訊軟體 APP 資料庫檔案比較表

常用通訊軟體APP資料庫檔案比較表				
				編碼
LINE	名稱	Talk.sqlite	Naver_line	UTF-8
	路徑	/private/var/mobile/Applications/jp.naver.line/Documents/Talk.sqlite /Applications/group.com.linecorp.line/Library/Application Support/Messages/Line.sqlite	/Root/data/jp.naver.line.android/databases/naver_line	
WeChat	名稱	MM.sqlite	EnMicroMsg.db	UTF-8
	路徑	/private/var/mobile/Application/com.tencent.xin/Documents/<WeChat_ID_MD5>/DB/MM.sqlite	/Root/data/com.tencent.mm/MicroMsg/<WeChat_ID_MD5>/MicroMsg.db	
	名稱	Main.db	Main.db	UTF-8
	路徑	/private/var/mobile/Applications/com.skype.skype/Library/ApplicationSupport/Skype/<UserName>/main.db	/Root/data/com.skype.raider/files/<UserName>/main.db	
	名稱	ChatStorage.sqlite	msgstore.db	UTF-8
	路徑	/private/var/mobile/Applications/group.net.whatsapp.WhatsApp.shared/ChatStorage.sqlite	/Root/data/com.whatsapp/databases/msgstore.db	

SQLite 的結構，由”RootPage”、”Internal Page”、”Leaf Page”所組成，其中”Internal Page”又可以分為”Internal Index Page”及”Internal Table Page”。而”Leaf Page”又可分為”Leaf Index Page”及”Leaf Table Page”。儲存在 SQLite Database 的內容、訊息(包含刪除資料)實際上就是儲存在 LeafTable Page 中。(詳見下圖 2)

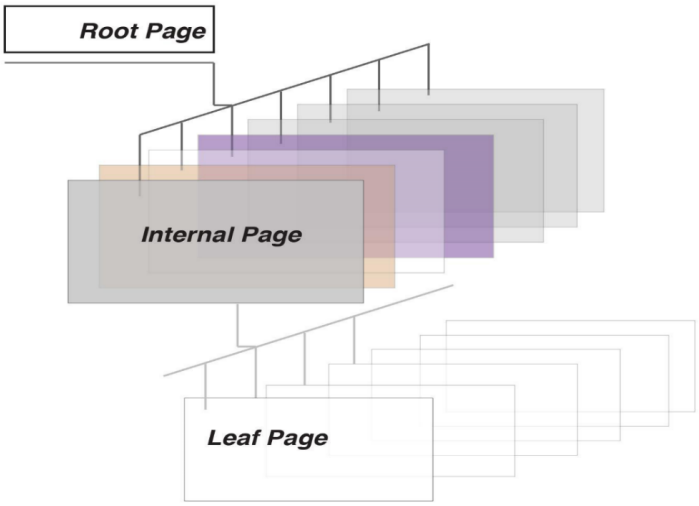


圖 2：SQLite Databse Page Structure

下圖 3 為”RootPage”的 header，第 16 個 byte 開始長度 2bytes 代表每一個 Page 的大小。例如”\x10 \x00”代表 Page 大小為 4096 bytes，又依照第 1 點可知 SQLite 均為 Page 組成，可知 SQLite 的檔案大小即為 4096 bytes 的倍數。0-4095 bytes 為第 1 個 Page，4096-8191 為第 2 個 Page 以此類推。

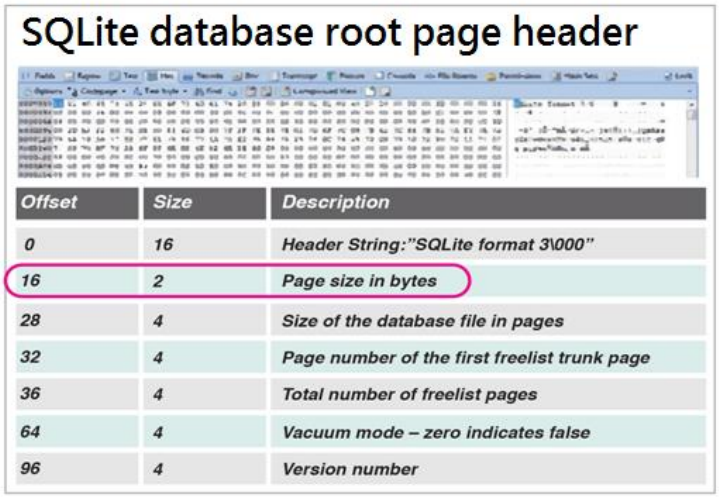


圖 3：SQLite Database root page header

除了”RootPage”外，各種 Page 的特徵值如下圖 4 所示，存放內容及訊息的 Leaf Table Page 的特徵值為\x0D。綜合上述條件，可以找到每一個 Page 起始的位址，而此位址只要是\x0D 開頭就有可能存放著刪除的訊息。

Page flag	Description
\x00	Freelist page
\x02	Interior index b-tree page
\x05	Interior table b-tree page
\x0A	Leaf index b-tree page
\x0D	Leaf table b-tree page

圖 4：SQLite database Leaf & Interior page header

3.2 Deleted data area

以硬碟為例，資料刪除後會移動到 Unallocated Space，SQLite 適用同樣的觀念但是多了一個 freeblock 區塊。故在 SQLite 內，刪除的資料會存在 Unallocated Space 以及 freeblock 兩個位置，必須將這兩個位置都檢視過才能回復所有刪除的資料。下圖 5 為”Leaf Table Page”的結構，其中只有看到 Unallocated Space，而

freeblock 實際上是位於 Cell content area 中。

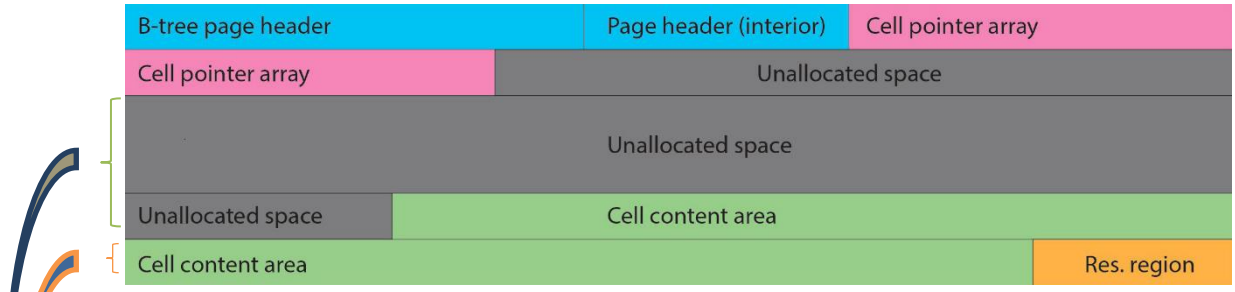


圖 5：Leaf Table Page structure

詳細說明如下圖 6 所示，第 2 個 freeblock 的意思為將此位置之 content area 的訊息刪除，而此位置前後的訊息都還存在的時候，此時就會變成 freeblock 區域。

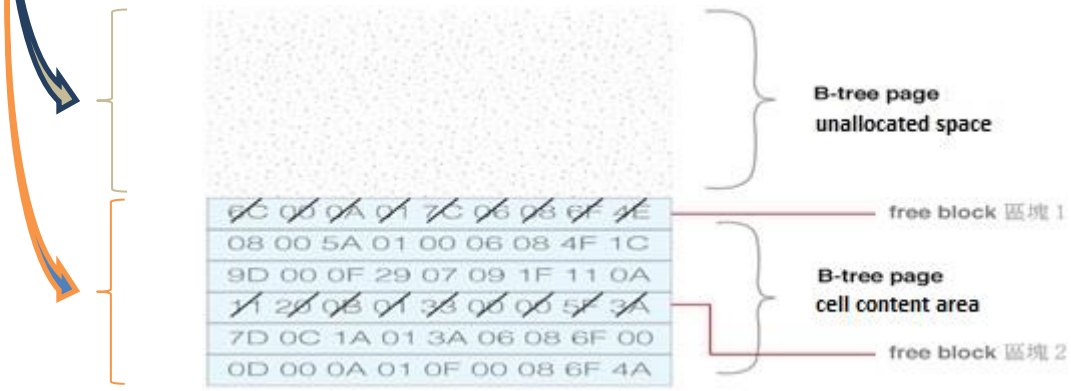


圖 6：SQLite database deleted data sturcture illustration

各種 Page 都有其 header 來代表，如下圖 7，第 0 個 byte “\x0D”代表此 Page 為 Leaf Table Page。第 1-2 byte 代表第一個 freeblock 位置，此位置就是儲存刪除訊息的位置之一。第 3-4 byte 代表此 Page 內現有訊息之數量，不包含刪除的訊息。第 5-6 byte 代表 Cell Content Area 起始的位置，配合圖 5 可以發現 Cell Content Area 上方即為 Unallocated Space，所以從此位置往上即可找到另一個儲存刪除訊息的位置。

Hex	Decimal	Description
0D	13	Page type - Leaf table b-tree
05 05	1285	Two byte integer - start of first freeblock
02 10	528	Two byte integer - number of cells
05 14	1300	Two byte integer - start of cell content area

圖 7：SQLite Database Leaf page header

因此如果要找到所有的 freeblock 的位置，除了第一個 freeblock 起始位置外還需要兩個資訊; freeblock 的大小及下一個 freeblock 的起始位置。這兩個資訊都可以從第一個 freeblock 位置的前 4 個 byte 找到。第 0-1 byte 代表下一個 freeblock 的位置，若為\x00\x00 則代表沒有下一個 freeblock。第 2-3 byte 代表此 freeblock 之大小，也代表 freeblock 的區塊。

每一個區塊中記錄著 Table 裡面所建立的欄位資訊內容，其中一個欄位就是儲存訊息的內容，故必須了解此區塊中每個 byte 所代表的意義才能找到刪除的訊息。

圖 8 代表 chat_history 這個 Table 中所創建出來的欄位名稱及型態。如第一個欄位名稱是"id"，型態為"Integer"，第二個欄位名稱為"server_id"，型態為"text"等以此類推。存放訊息的欄位名稱則是"content"，型態為"text"。特別要注意的是欄位名稱是創建者自行命名，不是所有 SQLite 文件存放訊息的欄位名稱都是"content"。

```
tory CREATE TABLE chat_history (
id INTEGER PRIMARY KEY AUTOINCRE
MENT,server_id TEXT,type INTEGER
,chat_id TEXT,from_mid TEXT,cont
ent TEXT,created_time TEXT,deliv
ered_time TEXT,status INTEGER,se
nt_count INTEGER,read_count INTE
GER,location_name TEXT,location_
address TEXT,location_phone TEXT
,location_latitude INTEGER,locat
ion_longitude INTEGER,attachemen
t_image INTEGER,attachement_imag
e_height INTEGER,attachement_ima
ge_width INTEGER,attachement_ima
ge_size INTEGER,attachement_type
 INTEGER,attachement_local_uri T
EXT,parameter TEXT)t    9#    inde
```

圖 8：各欄位之名稱及其型態

以 Leaf Table Page header `0D 0B 0E 00 0A 0A 93 00` 為例，第 1-2byte "\x0B\x0E"代表第 1 個 freeblock 起始位置，轉換成十進位後，第一個 freeblock 的起始位置在此 Page 起算的第 2830byte 的位置，如圖 9。

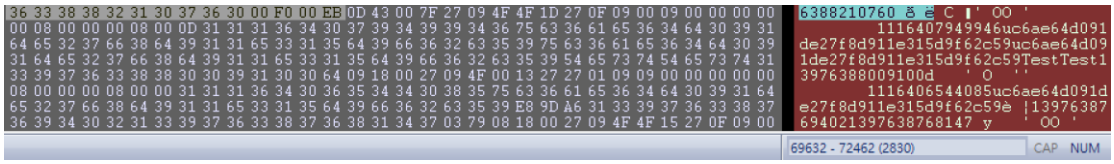


圖 9：此 page 第 2830byte 位置(代表第一個 freeblock 起始位置)

如圖 10 所示，第一個 freeblock header 從\x0D\x43\x00\x7F 開始，\x0D\x43 代表下一個 freeblock 的起始位置，\x00\x7F 代表此區塊的大小為 127bytes，至此可定位出第一個 freeblock 區塊的大小，但實際上代表欄位的只有 123bytes，原因是要扣掉最前面 4bytes 的資訊，故此 123bytes 會包含第一個 freeblock 所有欄

位的資訊。

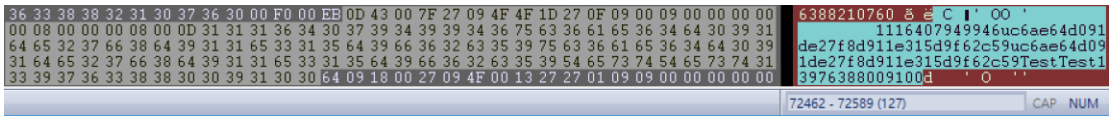


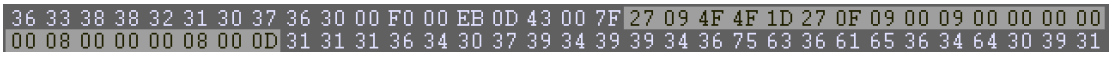
圖 10：第一個 freeblock 區塊大小

表 2：根據圖 8 整理之欄位名稱，共有 23 個欄位。

No.	Name	No.	Name	No.	Name
1	id	11	read_count	21	attachment_type
2	server_id	12	location_name	22	attachment_local_uri
3	type	13	location_address	23	parameter
4	chat_id	14	location_phone		
5	from_mid	15	location_latitude		
6	content	16	location_longitude		
7	created_time	17	attachment_image		
8	delivered_time	18	attachment_image_height		
9	status	19	attachment_image_width		
10	sent_count	20	attachment_image_size		

3.3 刪除資料之回復還原

在 SQLite 中 1 個欄位會使用 1 個 byte 來表示狀態，故應有 23bytes 各代表 23 個欄位，但因刪除之 freeblock 區塊並沒有”id”或次序的概念所以不會有代表”id” 的 byte，所以此例總共有 22bytes 來代表 22 個欄位，如下圖反白區域。



“\x0D\x43”代表下一個 freeblock 的起始位置，”\x00\x7F”代表此區塊的大小為 127bytes，後續部分就是一個 byte 代表一個欄位狀態。所以第一個 byte 為”\x27” 代表是”server_id”，”\x27”十進位為 39，而 39 所代表的意義請參考表 3，

表 3：Byte 值對照表(SQLite 官方網站公布)

Serial Type Codes Of The Record Format

Serial Type	Content Size	Meaning
0	0	Value is a NULL.
1	1	Value is an 8-bit twos-complement integer.
2	2	Value is a big-endian 16-bit twos-complement integer.
3	3	Value is a big-endian 24-bit twos-complement integer.
4	4	Value is a big-endian 32-bit twos-complement integer.
5	6	Value is a big-endian 48-bit twos-complement integer.
6	8	Value is a big-endian 64-bit twos-complement integer.
7	8	Value is a big-endian IEEE 754-2008 64-bit floating point number.
8	0	Value is the integer 0. (Only available for schema format 4 and higher.)
9	0	Value is the integer 1. (Only available for schema format 4 and higher.)
10,11		*Not used. Reserved for expansion.*
N≥12 and even	(N-12)/2	Value is a BLOB that is (N-12)/2 bytes in length.
N≥13 and odd	(N-13)/2	Value is a string in the text encoding and (N-13)/2 bytes in length. The nul terminator is not stored.

此表為 SQLite 官網所公佈之格式。所以 39 來查表，可以發現 39>13 且為奇數，代表需要將(39-13)/2=13，所以最後的意思就是”server_id”用 13 個 bytes 來表示其值且形式為”string”。而這 13 個 bytes 就是接在代表欄位的 22bytes 之後。所以此例代表”server_id”的 13bytes 就是：

此為 ASCII 編碼，轉換成十進位就是1116407949946，所以 1116407949946 即為此筆刪除資料的 server_id。

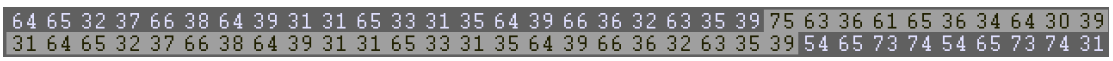
下一個 byte 為”\x09”代表”type”欄位，同樣查表，”\x09”代表其值為”1”但是不需要占用後方 byte 來表示。

下一個 byte 為”\x4F”代表”chat_id”欄位，”\x4F”的十進位為 79，79>13 且為奇數，(79-13)/2=33，”chat_id”欄位用 33bytes 來表示其值，此時 33bytes 就要從代表 server_id 的 13 個 bytes 後面開始算。所以就是下圖反白的部分：

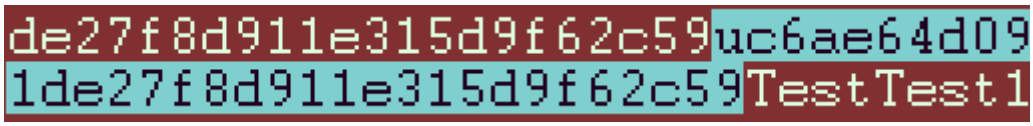
同樣為 ASCII 編碼，轉換成十進位後其值為下圖反白的部分：

所以 uc6ae64d091de81f8d911e315d9f62c59 就是此刪除訊息的 chat_id。

下一個 byte 同樣為”\x4F”，代表的是”from_mid”欄位。”\x4F”十進位為 79，79>13 且為奇數，(79-13)/2=33，33bytes 來表示”from_mid”欄位的值，同上所述此 33bytes 從”chat_id”的 33 個 bytes 後面開始算。即為下圖反白部分：



同樣為 ASCII 編碼，轉換成十進位為下圖反白的部分：

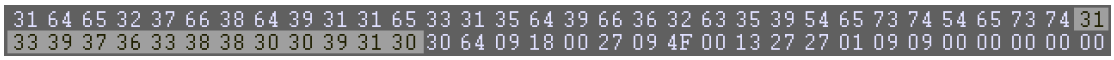


所以 uc6ae64d091de81f8d911e315d9f62c59 就是此刪除訊息的 from_mid。下一個 byte 為”\x1D”，代表”content”欄位。”\x1D”轉成十進位為 29，29>13，(29-13)/2=8，故用 8bytes 來表示”content”欄位的值。8bytes 從代表 from_mid”的 33 個 bytes 後面開始算。所以就是反白的部分：

，訊息內容編碼為 UTF-8，轉換成 UTF-8 編碼後其值為

，所以此”TestTest”就是被刪除的訊息，也就是所要還原的目標之一。

下一個 byte 為”\x27”，代表”created_time”欄位。”\x27”轉成十進位為 39，39>13 且為奇數，(39-13)/2=13，用 13bytes 來表示”created_time”欄位的值。此時 13bytes 從代表”content”的 8 個 bytes 後起算。為下圖反白的部分



同樣為 ASCII 編碼，轉換成十進位為下圖反白的部分

所以 1397638800910 就是此筆訊息建立的時間，要注意的是此 13 碼數字為 timestamp 形式，必須經過轉換才能轉成我們看得懂的格式。經轉換後此筆訊息建立的時間為 2014/04/16 9:00am (UTC)。

下一個 byte 為”\x0F”，代表”delivered_time”欄位。”\x0F”轉成十進位為 15，15>13，(15-13)/2=1，用 1 byte 來表示”delivered_time”欄位的值。此時 1 byte 從代表”created_time”的 13 個 bytes 後起算。為下圖反白的部分

同樣為 ASCII 編碼，轉換成十進位如下圖反白的部分，其值為 0。

其他的 byte 所代表的值就如同上述查表、對應、轉換等步驟後即可依序找出，上述分析的結果整理如下表 4

表 4：各 byte 值對應之 Value

No.	Name	Value	No.	Name	Value
1	id	null	13	location_address	null
2	server_id	1116407949946	14	location_phone	null
3	type	1	15	location_latitude	null
4	chat_id	uc6ae64d091de81f8d911e315d9f62c59	16	location_longitude	null
5	from_mid	uc6ae64d091de81f8d911e315d9f62c59	17	attachment_image	0
6	content	TestTest	18	attachment_image_height	null
7	created_time	1397638800910	19	attachment_image_width	null
8	delivered_time	0	20	attachment_image_size	null
9	status	1	21	attachment_type	0
10	sent_count	null	22	attachment_local_uri	null
11	read_count	1	23	parameter	null
12	location_name	null			

* null 與 0 代表之意義不同

最後只要將所有 freeblock 以及 UnallocatedSpace 依上述相同方式進行分析，即可將此 SQLite 檔案內之刪除訊息全數還原。

肆、結論

本篇所提到還原刪除訊息之方法不僅可用於”Line”，對於使用 SQLite Database 儲存資料的 App 軟體均適用其原理。但需要注意是必須先找出該 App 儲存資料的檔案才能進行分析，否則分析到錯誤的貯存位置及檔案將徒勞無功。其次以上案例只有提到”chat_id”及其值，並未明確提到”帳號”，必須尋找其他能明確定義”chat_id”及其”帳號”之 Table，如此才能賦予訊息”意義”，否則還原的就只是片面資訊無法對應收送方的對象為何。最後本篇主要應用於手機鑑識中提供執法人員一種方式及手段來還原被刪除之對話紀錄，對於資料救援領域及惡意程式 APP 上的 Database 操作行為亦可加以利用。

參考文獻

[1] Douglas Comer, *The Ubiquitous B-Tree*, Computer Science Department, Purdue University, 1979.

[2] Martin Westman, "Analysing freepages in SQLite databases," *Microsystem*, 2014.

[3] SQLite website, "SQLite file format, " https://www.SQLite.org/fileformat2.html

[4] 王随刚、吴莎莎、 李昂， "基于SQLite3的Android手机数据恢复技术的研究"， *北京锐安科技有限公司*，2012。