情報保護學會論文誌
第 22 卷 第 1 號, 2012. 2

삭제된 휴대폰 음성 데이터 복원 방법론*

김 상 대,† 변 근 덕, 이 상 진‡
고려대학교 정보경영공학전문대학원

Carving deleted voice data in mobile*

Sangdae Kim,† Keunduck Byun, Sangjin Lee‡
Korea University, Graduate School of Information Management and Security

요 약

휴대폰에서 사용하는 대화내용을 녹음하거나 음성으로 메모를 남기는 경우가 있는데 범죄의 은폐나 사용자의 실수로 중요한 음성 데이터를 삭제하는 경우가 있다. 음성 데이터는 어떤 사실에 대한 증거로써 영향력이 강하기 때문에 포렌식 조사를 위해서도 삭제된 음성 데이터를 복구해야 한다. 데이터가 조각나기 쉬운 플래시 메모리에 데이터를 저장하는 휴대폰의 특성상 음성 데이터를 복구하기 어렵다. 하지만 음성 데이터를 특정할 수 있는 패턴이 있다면 이 패턴으로 이미지를 조사하여 음성 데이터를 일정 이상 복원할 수 있다. 음성 데이터에는 여러 종류가 있고, 본 논문에서는 퀄컴의 QCP 파일 포맷에서 사용하는 EVRC, AMR 코덱에 대하여 데이터를 복구할 수 있는 방안을 제안한다.

ABSTRACT

People leave voicemails or record phone conversations in their daily cell phone use. Sometimes important voice data is deleted by the user accidently, or purposely to cover up criminal activity. In these cases, deleted voice data must be able to be recovered for forensics, since the voice data can be used as evidence in a criminal case. Because cell phones store data that is easily fragmented in flash memory, voice data recovery is very difficult. However, if there are identifiable patterns for the deleted voice data, we can recover a significant amount of it by researching images of it. There are several types of voice data, such as QCP, AMR, MP4, etc.. This study researches the data recovery solutions for EVRC codec and AMR codec in QCP file, Qualcumm's voice data format in cell phone.

Keywords: Mobile Data Carving, Digital Forensics

I. 서 론

스마트폰과 다양한 기능을 가진 피처폰이 보급되면서 휴대폰으로 통화내용을 녹음하거나 음성 녹음 기능으로 필요한 메모를 남기는 경우가 늘어나고 있다. 녹음된 음성 내용은 법정 대립 시 중요한 단서를 제공하기도 하므로 중요한 음성 데이터를 실수로 삭제하거나 고의로 삭제할 경우 이를 복원할 필요가 있다. 휴대폰에서 사용하는 플래시 메모리는 매체 특성으로 인해 현재 많이 사용하는 카빙 방법을 적용하기 어렵다. 따라서 휴대폰에서 음성 데이터를 복원하기 위해서는 음성 파일 및 코덱의 특징에 따라 복원을 수행해야 한다.

휴대폰의 음성 녹음 프로그램들이 사용하는 음성 코덱으로는 EVRC, SMV, QCELP, AMR 등이 있고 국내 피처폰의 경우 AMR, EVRC, SMV, QCELP를 지원하는 QCP 파일 포맷과 AMR 파일 포맷을 많이 사용하고 있으며, 스마트폰의 경우 사용

접수일(2011년 3월 23일), 게재확정일(2012년 2월 2일)
* 본 연구는 한국연구재단을 통해 교육과학기술부의 바이오 연구개발사업으로부터 지원받아 수행되었습니다. (20100020634)
† 주저자, freekuku@korea.ac.kr
‡ 교신저자, sangjin@korea.ac.kr

하는 응용 프로그램에 따라 다양한 파일 포맷과 코덱을 지원하고 있다. 음성 데이터는 파일의 포맷과 코덱에 따라 인식 가능한 패턴을 활용하여 음성을 복구할 수 있다.

본 논문에서는 음성 파일 포맷 중 퀄컴에서 제안한 QCP 파일 포맷의 특징을 이용하여 물리 이미지에서 음성을 복구하는 방법을 제안한다.

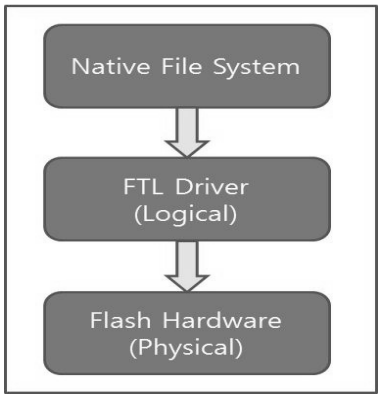
II. 관련 연구

데이터 카빙에 대한 연구에서는 Header, Footer, 파일 크기 등의 데이터를 기준으로 연속된 데이터를 파일로 만드는 방법이 많이 사용되었고, 최근 조각난 파일에 대하여 파일 포맷의 특성을 이용하여 파일의 적합성을 검사하여 조각난 파일을 재구성하는 연구가 있었다[5]. 또 가장 최근 연구로는 조각난 JPEG 그림 파일에 대한 카빙 방법이 제시 되었다. JPEG 파일은 파일 포맷의 특성이 아닌 그림의 경계선에서 나타나는 특징, 즉 데이터 자체의 특성을 이용하여 그림의 경계면을 검사하고 유효한 조각을 찾는 연구가 있었다[6].

III. 플래시 메모리의 특징

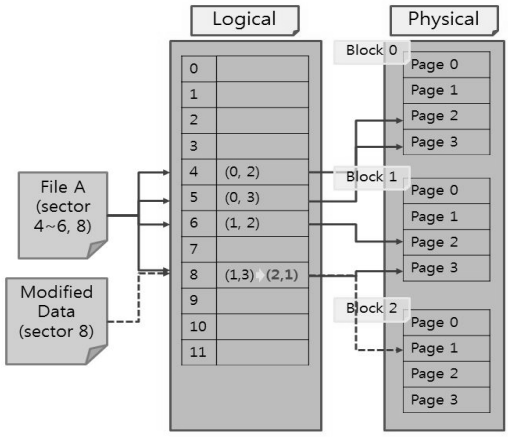
플래시 메모리는 전원이 없어도 데이터가 유지되는 저장 매체로써 현재 작은 크기와 빠른 속도 때문에 많은 곳에서 쓰이고 있으며, 특히 모바일 장비들에서 데이터를 저장하기 위해 대부분 플래시 메모리를 사용하고 있다. 플래시 메모리에서는 어떤 데이터를 쓰기 위해서 이전 데이터를 덮어 쓰는 것이 아닌 특정 크기의 블록을 먼저 지우고 그 영역에 데이터를 기록한다. 이때 데이터를 읽고 쓰는 단위보다 지우는 블록의 크기가 크고, 지우지 않으면 다른 데이터를 다시 쓸 수 없기 때문에 낭비되는 공간이 많아진다. 그래서 빈번하게 데이터의 위치를 옮겨 데이터의 공간 활용성을 높인다. 더욱이 지우는 동작은 횟수에 제약이 있기 때문에 저장 장치로써 수명을 유지시키기 위해 플래시 메모리는 매체의 모든 영역을 고르게 사용한다. 이를 위해 wear-leveling과 garbage collection을 사용하며, 이는 플래시 메모리의 컨트롤러에서 수행한다.

wear-leveling은 데이터가 저장되는 셀이 고르게 사용되도록 하는 작업으로 플래시 메모리의 수명을 연장시킨다. 이를 위해 〔그림 1〕과 같이 파일 시스템과 물리적인 저장 공간 사이에 추가적인 단계를 두어 데



〔그림 1〕 플래시 메모리의 데이터 관리 구조

이터를 쓰는 공간을 결정한다. 이런 작업으로 인해 〔그림 2〕와 같이 논리적으로는 연속으로 기록된 데이터가 실제 물리적 저장 공간에서는 비연속적일 수 있다. 또 지우는 횟수를 줄이기 위하여 어떤 데이터가 수정되었을 경우 이전 데이터의 위치에 수정된 데이터로 바꾸는 대신 다른 공간에 수정된 데이터를 쓴다. 예를 들어 〔그림 2〕와 같이 파일 A가 파일 시스템 상에서는 4번 페이지에서 6번 페이지까지 그리고 8번 페이지를 사용하고 있을 때 FTL은 물리적으로 저장되어 있는 곳으로 맵핑한다. 논리적으로 4번 페이지부터 6번 페이지까지 연속적이지만 물리적으로 비연속적이다. 그리고 파일 A에 데이터를 수정하였을 경우 〔그림 2〕에서와 같이 논리적으로는 저장 위치가 수정된 것이 없다. 물리적으로는 Block1-Page3에서 Block2-Page1로 테이블 맵핑만 바꿀 뿐 이전 데이터를 직접 수정하지 않는다. 하지만 이전에 Block2에



〔그림 2〕 플래시 메모리의 wear-leveling 구조

저장되었던 정보들을 모두 지우고 쓰기 때문에 사용하지 않는 Page0, Page2, Page3의 정보들도 지워진다.

garbage collection은 가용 저장 공간을 확보하기 위한 작업이다[4]. 디스크 조각 모음과 같이 사용하지 않는 블록으로 가용 페이지들을 재배치하여 블록 내에 빈 페이지를 최소화한다.

wear-leveling과 garbage collection 때문에 플래시 메모리에서는 데이터를 쓸 때 연속적으로 데이터를 쓰는 경우가 드물고 연속적으로 써졌다 하더라도 garbage collection이 발생할 경우 데이터의 위치가 바뀔 수 있다. 따라서 일반적인 데이터 카빙 방법을 직접 적용하기 어렵다. 또 데이터를 지우거나 수정했을 경우 삭제나 수정하기 전의 데이터가 그대로 남아 있을 가능성이 높아 디지털 포렌식 수사를 위한 복원 작업이 반드시 필요하다.

IV. 음성 파일

4.1 음성 파일의 종류

음성 파일은 마이크로 전달된 아날로그 소리를 디지털로 바꾸어 저장한다. 이렇게 디지털 신호로 저장할 때 압축하는 방식에 따라 음질이 결정되고 이런 압축 방법을 코덱이라 한다. 휴대폰에서 사용하는 음성 코덱에는 EVRC, QCELP, AMR, SMV 등이 있다.

EVRC 방식은 8비트로 표현되는 샘플을 구하여 20ms 마다 압축하고 압축된 데이터를 1개 프레임으로 표현하며 가변 프레임을 사용한다. full rate 일 경우 171bits, half rate 경우 80bits, eight rate 경우 16bits로 표현한다. 같은 방식으로 SMV는 full(170bits), half(80bits), quarter(40bits), eighth(16bits)를 사용한다[2][3]. AMR 방식은 GSM 계열에서 주로 쓰는 방법으로 mode와 rate를 구분하고 두 값의 조합에 따라 코딩 방법을 다르게 적용한다[1].

4.2 휴대폰 음성 데이터 저장 방식

음성 파일은 음성 신호를 해석하여 저장하는 데이터 부분과 전체 음성 데이터를 어떻게 저장할지를 결정하는 컨테이너 부분으로 나뉜다. 컨테이너는 음성 데이터에 대한 메타 정보들을 저장하고 음성 데이터는 지정된 코덱으로 압축, 저장한다.

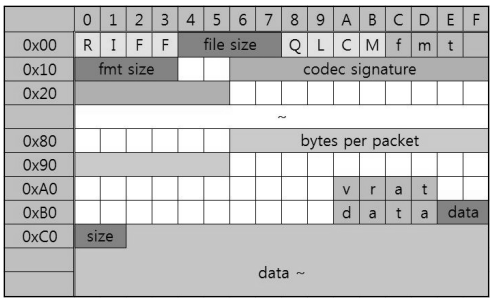
우리나라에서 사용되는 피쳐폰의 경우 퀄컴에서 제안한 QCP 파일 포맷을 많이 사용하며 GSM 계열의 휴대폰은 AMR 파일 포맷을 사용한다. 우리나라는 3세대 휴대폰 중 SKT와 KT는 GSM 계열인 WCDMA를 LGT는 CDMA 계열인 CDMA Rev.A를 쓰고 있다. 2세대 때는 모두 CDMA를 쓰고 있었기 때문에 퀄컴의 포맷인 QCP 파일 포맷을 많이 사용하고 3세대에서는 QCP 또는 AMR 파일 포맷을 컨테이너로 사용한다. 스마트폰의 경우 iPhone은 MPEG4를 기본 포맷으로 사용하고 안드로이드는 AMR, 3GP 등 사용하는 응용 프로그램에 따라 다양한 파일 포맷을 사용한다.

4.3 QCP 파일 포맷

QCP 파일 포맷은 퀄컴에서 제안한 포맷으로 Resource Interchange File Format(RIFF)을 기반으로 구성되어 있으며 음성 녹음을 지원하는 국내 피쳐폰에서 주로 사용한다. 자체 코덱은 없지만 다른 음성 코덱을 지원할 수 있는 컨테이너 타입으로, 지원하는 코덱은 QLEP13K, AMR, EVRC, SMV가 있다. QCP 파일 포맷은 음성을 재생하는데 필요한 정보를 저장하는 헤더와 음성 데이터를 저장하는 데이터 영역으로 구분할 수 있다.

4.3.1 QCP 메타 데이터

QCP 포맷의 헤더는 [그림 3]과 같다. 반드시 필요한 메타 데이터는 "fmt ", "vrat", "data"가 있으며 옵션 메타 데이터로 "labl", "offs", "cnfg", "text"가 있다. 이 중 "labl"과 "offs"의 위치는 "vrat"와 "data"



[그림 3] QCP 파일의 헤더 포맷

[그림 4] QCP 파일의 메타 데이터 구성

〔표 1〕 QCP 파일에서 사용하는 코덱 ID

Codec Name	Codec Value
QCELP-13K	0x416D7F5E15B1D011BA9100805FB4B97E
	0x426D7F5E15B1D011BA9100805FB4B97E
EVRC	0x8D4D89E67690B54691EF736A5100CEB4
SMV	0x752B7C8D97A749ED985ED53C8CC75F84
AMR	0x53E0A86A4F4746BD8AFAACF2328273BD

〔그림 5〕 EVRC 코덱의 "bytes per packet"

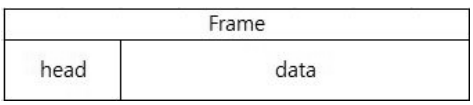
사이로 고정되어 있지만 cnfg와 text는 위치에 제약이 없다.

"RIFF" 뒤에 나오는 4바이트는 "QLCM"부터 파일 끝까지의 크기가 저장되어 있다. "fmt"에 있는 "codec signature"는 코덱의 종류에 따른 고유 ID를 저장하고 있어 사용된 코덱의 종류를 알 수 있다. 〔표 1〕은 QCP 파일에서 사용하는 코덱의 ID이다.

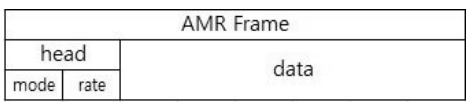
"bytes per packet"은 프레임 헤더에 따른 데이터 크기를 저장한다. 프레임 헤더에 따른 데이터의 크기는 〔그림 5〕와 같다. 2바이트 묶음으로 정보를 표시하고 앞의 데이터가 헤더를 제외한 프레임의 크기이고 뒤의 것은 프레임에서 헤더의 값이다. 예를 들어 프레임의 헤더 값이 0x04라면 그 뒤로 0x16바이트의 음성 데이터가 나온다. "bytes per packet" 항목은 QCELP13K, EVRC, SMV 코덱을 쓰는 경우에만 사용되며 AMR 코덱을 사용하는 경우에는 AMR 고유의 방법을 이용한다.

4.3.2 QCP 파일 포맷의 음성 데이터

data 영역에서는 "data" 식별자와 4바이트의 data 영역 크기 값 다음부터 음성 데이터가 나온다.



〔그림 6〕 QCP 파일의 프레임 구성



〔그림 7〕 QCP 파일 포맷에서 AMR 코덱을 사용할 경우 프레임 구성

음성 데이터는 각각의 프레임으로 구성되어 있으며 〔그림 6〕과 같이 1바이트의 프레임 헤더와 그에 따른 가변 크기의 데이터로 구성되고, 프레임 데이터는 QCP 메타 데이터에서 지정한 코덱으로 저장한다.

QCELP13K, EVRC, SMV 코덱을 사용하는 경우에는 "bytes per packet" 메타 데이터를 통해 프레임 헤더에 따른 프레임 데이터 길이를 알 수 있다.

AMR 코덱은 앞의 코덱과는 다르다. 〔그림 7〕과 같이 1바이트의 프레임 헤더가 4비트 씩 나뉘어 AMR frame mode와 AMR frame rate로 사용된다. AMR frame mode와 AMR frame rate의 조합에 따라 〔표 2〕의 값을 참고하면 프레임 데이터는 0~31바이트까지 생성된다. AMR frame mode가 7인 경우 전달할 음성이 없는 경우로써 이때는 프레임 데이터의 크기가 0이므로 프레임 헤더만 나타난다. AMR frame mode가 0인 경우는 정상적인 음성으로 AMR frame rate의 값에 따라 프레임 데이터의 길이가 결정된다.

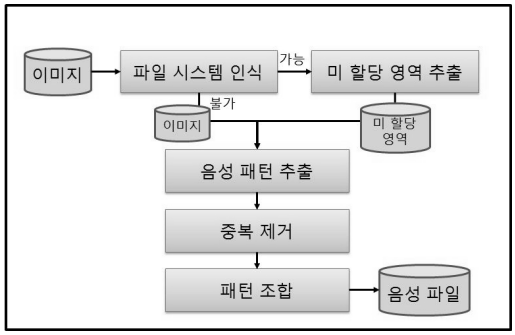
V. 삭제된 음성 파일 카빙

음성 데이터는 프레임으로 이루어져 있으므로 프레임 헤더에 따른 데이터 길이를 패턴으로 보고 같은 규칙을 따르는 모든 데이터를 수집하여 재조합하면 음성 파일을 복원할 수 있다.

〔표 2〕 AMR 코덱의 "bytes per packet"

Mode	Data Length
0	rate
1	-
2	-
3	5
4	5
5	5
6	5
7	0

Rate	Data Length
1	12
2	13
3	15
4	17
5	19
6	20
7	26
8	31



〔그림 8〕 음성 데이터 카빙 과정

5.1 페이지 크기 검색

플래시 메모리는 페이지 단위로 읽기와 쓰기를 수행하므로 삭제된 데이터 조각을 찾기 위해서는 페이지 크기를 알아낼 필요가 있다. 만약 페이지 단위보다 더 큰 단위로 데이터를 검색한다면 음성 데이터와 그렇지 않은 데이터가 섞여 있을 수 있다. 반대로 페이지 단위보다 작은 데이터 크기를 기준으로 검색 한다면 추출한 데이터가 지나치게 조각난 상태로 수집되어 데이터를 조합할 때 다른 음성과 섞이는 현상이 심해진다. 하지만 대상 플래시 메모리의 쓰기 단위를 정확하게 알 수 없을 때는 최소 단위를 고려하여 최대한 많은 정보를 찾아야 한다.

5.2 미 할당 영역 추출

미 할당 영역은 파일 시스템에서 명시적으로 사용하고 있는 공간을 제외한 나머지 영역이다. 삭제된 음성 데이터는 미 할당 영역에 데이터가 남아 있으므로 미 할당 영역만을 검사하여 효율성을 높일 수 있다.

먼저 파일 시스템에서 사용하는 공간을 제외한 후 파일 시스템 데이터와 남은 공간의 데이터를 비교하여 같은 데이터를 제외한다. 이 과정을 거친 후 남은 데이터가 미 할당 영역이다.

미 할당 영역을 추려낼 수 있다면 미 할당 영역에서 음성 패턴을 추출하고 파일 시스템을 알 수 없어 미 할당 영역만을 추려낼 수 없다면 전체 이미지를 대상으로 음성 패턴 추출을 한다.

5.3 음성 패턴 추출

파일의 헤더는 메타 데이터를 포함하고 있으며 음성 데이터는 프레임의 연속으로 구성되어 있다. 따라서 파일의 헤더와 데이터는 저장되는 방식이 다르므로 별도의 과정으로 추출을 시도해야 한다.

5.3.1 음성 파일 헤더 추출

플래시 메모리는 페이지 단위에 맞추어 데이터를 쓰기 때문에 파일 헤더의 식별자인 "RIFF"를 페이지의 시작 위치에서 찾는다. "RIFF"만으로는 다른 멀티미디어 파일 포맷과 구분할 수 없기 때문에 이 파일이 QCP 파일의 헤더인지 확실히 알기 위해서는 페이지 시작에서 0x08 위치에 있는 "QLCM" 시그니처를 확인하여 현재 파일이 QCP 파일인지 확인한다. 파일 헤더가 확인되었다면 파일 헤더에 있는 코덱 ID를 확인하고 물리 이미지에서 헤더가 발견된 오프셋 값을 기록한다.

5.3.2 음성 패턴 추출

음성 데이터의 프레임은 프레임 헤더와 프레임 데이터로 구분되고 헤더의 값에 따라 데이터의 길이가 다르다는 특징을 패턴으로 하여 구분한다.

음성 데이터를 페이지 단위로 자를 때 각각의 페이지 처음과 끝에 있는 프레임 데이터가 잘리게 되어 페이지의 시작점에서는 프레임 헤더보다 프레임 데이터가 먼저 나올 수 있다. 이는 앞의 어떤 페이지가 있고, 그 페이지의 마지막 프레임에서 모자라는 프레임 데이터가 현재 페이지에 있음을 의미한다. 각 페이지의 처음과 끝에 나타날 수 있는 음성 데이터의 패턴은 〔그림 9〕, 〔그림 10〕, 〔그림 11〕과 같이 구분할 수 있다. 〔그림 9〕를 보면 'Page1'의 마지막 프레임은 프레임 헤더 다음에 12바이트만이 동일한 페이지에 존재하고 나머지 4바이트는 다음 페이지에 존재한다. 〔그림 10〕을 참고하면 프레임 헤더의 위치를 고려하였을 때 'Page1'의 마지막 프레임이 온전하게 끝난 경우 다음 페이지 시작 위치에 프레임 헤더가 나온다. 〔그림 11〕은 프레임 헤더로 끝난 경우 프레임 헤더에서 필요한

〔그림 9〕 QCP 파일의 프레임 데이터

〔그림 10〕 프레임 데이터가 온전하게 끝나는 경우

〔그림 11〕 프레임 헤더로 페이지가 끝나는 경우

프레임 데이터가 전부 다음 페이지에 나온다. 그래서 페이지의 시작에서 프레임 헤더를 찾을 때, 가능한 최대 프레임 데이터 크기에 프레임 헤더 크기를 합하여 검사한다.

프레임 헤더를 찾았다면 해당 페이지가 음성 데이터라 가정하고 규칙에 맞는지 검사한다. 〔그림 9〕을 참조하여 범위 내에 프레임 헤더로 사용할 수 있는 값(H 표시)이 있다면 프레임 헤더 앞에 나오는 데이터를 프레임 데이터로 보고 그 값을 프레임 데이터의 나머지 값(시작 크기)으로 기록한 후 페이지 끝까지 지정된 프레임 헤더와 헤더에 따른 데이터가 적합한지 검사한다. 페이지의 마지막 프레임의 경우 프레임 헤더 값에 따른 프레임 데이터가 모두 나오지 않을 수 있다. 모자라는 데이터 값을 다음 페이지에 나올 프레임 데이터 값(필요 크기)으로 기록한다. 이렇게 음성 데이터로 판단되는 페이지를 이미지에서 모두 추출한 다음 각 페이지들을 연결하기 위해 필요한 시작 크기와 필요 크기 그리고 페이지의 오프셋 등을 저장하여 패턴 파일들을 조합하기 위한 정보로 활용한다.

음성 파일의 마지막 페이지의 경우 페이지 끝까지 사용하지 않을 수 있다. 파일 헤더가 있는 경우 헤더에 표시된 크기 정보를 바탕으로 정확한 크기를 예측해 볼 수 있지만 옵션 메타 데이터의 유무나 파일 헤더가 없어진 경우를 고려해 봤을 때 한 개 페이지 전체가 패턴과 일치하는 것만 찾는 것이 효과적이다.

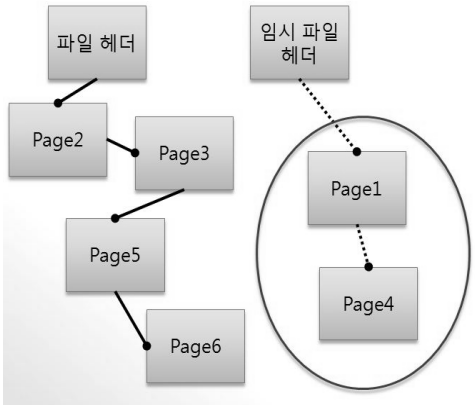
5.4 중복된 음성 데이터 제거

플래시 메모리의 특성 때문에 같은 데이터가 여러 페이지에 있는 경우가 있다. 이런 경우 중복 데이터를 처리하지 않고 음성을 조합할 경우 조합한 음성이 울림소리 같이 만들어지거나 같은 말을 반복하는 경우가 있다. 따라서 음성 데이터를 올바르게 복원하기 위해서 중복되는 음성 패턴을 제거해야 한다. 추출한 패턴 파일들은 페이지 단위로 뽑을 경우 그 페이지의 크기가 크지 않다면 직접 비교하는 것이 좋고, 비교해야 할 데이터의 크기가 크다면 해쉬 함수를 이용하여 해쉬 값을 구한 후 그 값을 비교하는 방법을 사용한다.

5.5 음성 파일 조합

파일 헤더를 기준으로 음성 데이터를 조합하는 방법과 모든 음성 데이터 조합에 대하여 파일 헤더를 붙이는 방법이 있다. 파일 헤더를 기준으로 음성 데이터를 조합할 경우 중간 부분이 누락되거나 다른 데이터가 연결되어 조합되지 않는 음성 데이터가 남을 수 있다. 음성 데이터의 모든 조합을 만들고 파일 헤더를 붙여서 음성 파일을 만들면 너무 많은 수의 음성 파일이 생긴다. 많은 음성 파일은 사람이 유효성을 검증해야 하는데 그렇게 하기에는 비효율적이다. 따라서 본 논문에서는 유효성 검사의 현실성을 고려하여 발견된 음성 파일의 헤더에 음성 패턴을 연결하는 방식을 연구하였다.

파일 헤더가 있는 경우 우선적으로 조합하고 그 후 남은 데이터는 임시 파일 헤더를 만들어 조합한다. 임시 파일 헤더는 QCP 파일에서 필요로 하는 필수 메타 데이터와 임의의 음성 프레임으로 작성한다. 〔그림 12〕를 보면 음성 파일 헤더에 맞는 음성 데이터를 먼저 조합한다. 6개의 페이지 중 4개 페이지가 음성 파



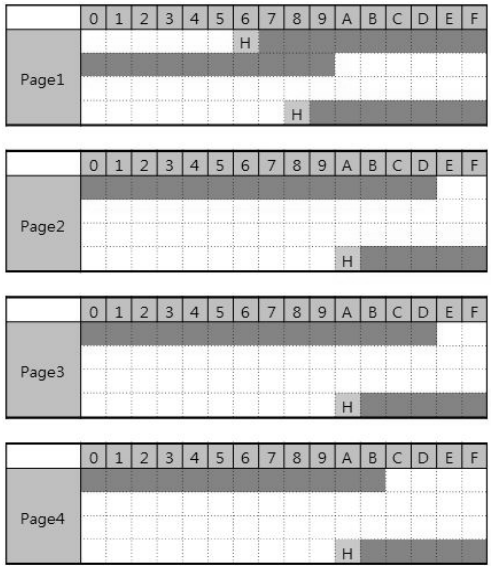
〔그림 12〕 추출한 데이터의 연결 예시

일 헤더와 조합되고 2개 페이지가 남는다. 남은 2개 페이지에도 음성 데이터가 있으므로 여기에 임시 파일 헤더를 붙여 음성 파일을 만든다.

임시 파일 헤더는 필수 메타 데이터와 음성 데이터로 구성한다. 음성 데이터에 들어갈 프레임은 소리가 없을 때 사용하는 프레임으로 채워 넣고 페이지 마지막에서는 다음에 올 음성 패턴에 맞게 임의의 프레임을 생성한다. AMR에서는 AMR frame mode가 0x7인 것을 사용하고 EVRC에서는 프레임 헤더가 0x01인 것을 사용하여 부적절한 음성이 추가되는 것을 방지한다.

음성 데이터들을 조합할 때는 현재 페이지의 필요 크기와 시작 크기가 같은 페이지를 찾는다. 그 다음부터 계속 필요 크기와 시작 크기를 비교하여 조건에 맞는 음성 데이터를 붙인다. 연결 가능한 음성 데이터가 없거나 음성 파일의 크기가 일정 이상으로 커지면 조합을 중지한다.

서로 다른 페이지를 연결할 때 사용할 수 있는 패턴의 수는 프레임의 최대 길이와 같다. 예를 들어 AMR 코덱을 사용하는 경우 32가지 패턴이 나온다. 512바이트 단위로 음성 데이터를 추출한다면 음성 데이터의 수에 비해 패턴이 너무 적기 때문에 연결할 수 있는 음성 데이터가 많이 나온다. 이 때는 물리 이미지의 오프셋 값을 기준으로 현재 페이지보다 오프셋 값이 크면서 가장 가까운 음성 데이터를 먼저 조합한다. 위 조건에 맞는 데이터가 없고, 남아 있는 데이터가 있다면 오프셋 값이 가장 작고 연결이 가능한 페이지를 찾아 연결한다. 이렇게 오프셋에 맞는 음성 데이터를 찾는 이유는 원본 데이터가 연속된 페이지로 존재할 경우 가장 좋은 결과를 얻을 수 있기 때문이다.

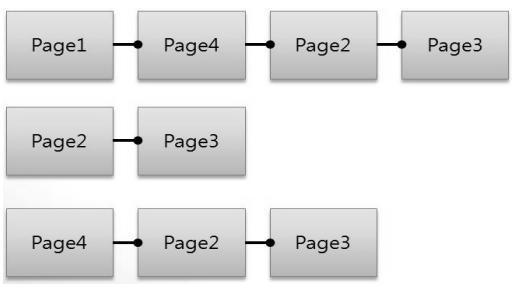


〔그림 13〕 추출한 데이터에 대한 예시

〔그림 13〕를 보면 이 음성 프레임은 프레임 헤더가 'H' 일 때 프레임 데이터 길이가 19이다. 'Page1'의 필요 크기가 12이다. 그래서 시작 크기가 12인 음성 데이터를 찾는다. 'Page2'와 'Page3'은 시작 크기가 14이므로 건너뛰게 되고 'Page4'의 시작 크기가 12이므로 'Page1' 뒤에 'Page4'를 붙인다. 'Page4' 뒤에는 더 이상 음성 데이터가 없으므로 앞에서부터 다시 검색한다. 'Page1'은 이미 사용했으므로 제외하고 'Page2'부터 검색한다. 'Page4'의 필요 크기와 'Page2'의 시작 크기가 일치하기 때문에 'Page4' 뒤에는 'Page2'를 붙인다. 같은 방법으로 'Page3'을 붙이고 더 이상 조합할 음성 데이터가 없으므로 종료한다. 이와 같은 방법으로 선행 음성 데이터에 따라 〔그림 14〕와 같은 음성 데이터 조합이 만들어진다.

VI. 결 론

본 논문에서는 삭제되어 저장 위치를 알 수 없고, 데이터가 조각나 카빙이 어려운 음성 파일에 대하여 음성 파일 포맷에서 데이터 프레임이 구분할 수 있는 특정한 규칙을 가지는 경우 이 것을 패턴으로 데이터를 추출하고 특정한 규칙이 부합되는 데이터를 연결하여 음성 파일을 카빙하는 방법을 제안하였다. 국내 폰을 대상으로 복구를 시도하였고 AMR 코덱으로 저장된 휴대폰에서는 파일 시스템 상의 음성이 4개가 있었고, 파일 헤더가 없는 1개의 음성 내용을 추가로 발견하였다. 음성은 10개의 조각으로 복원하였고, 각각 20~40초 정도의 대화 내용을 확인하였다. EVRC 코덱의 경우 파일 헤더는 발견하였지만 복원한 내용은 10초 정도로 짧은 경우가 있었다. 모든 내용을 완전하



〔그림 14〕 음성 데이터 연결 예시

게 복구하기는 힘들지만 대화의 일정부분은 내용을 들을 수 있었다.

여러 음성 파일이 저장, 삭제되면 음성 파일 조각들이 여기저기 흩어져 있고 이를 조합하게 될 때 여러 개의 다른 음성이 마치 하나의 음성 같이 붙어 청취 시 내용 판단에 영향을 준다. 하지만 휴대폰의 음성 데이터들은 대부분 어떤 정보를 기록하기 위하여 메모를 남기는 것이나 누군가와 통화를 녹음하기 위한 것이 대부분이고 이런 데이터는 1명 또는 2~3명 정도의 목소리가 기록된다. 또 하나의 페이지에 들어 있는 프레임 데이터에서 음성의 특징을 찾고 이를 바탕으로 서로 연결되는 페이지를 조합하여 음성 데이터 복원의 완성도를 높이는 연구가 필요하다.

참고문헌

〔1〕 Mandatory Speech Codec speech processing functions; Adaptive Multi-Rate (AMR) speech codec; Transcoding functions, 3GPP TS 26.090, December, 2009.

〔2〕 Enhanced Variable Rate Codec, Speech Service Option 3 for Wideband Spread Spectrum Digital Systems, 3GPP2 C.S0-014-0, 3GPP2 Version 2.0, January 2010

〔3〕 Selectable Mode Vocoder (SMV) Service Option for Wideband Spread Spectrum Communication Systems, 3GPP2 C.S00-30-0, 3GPP2 Version 3.0, January 2004

〔4〕 Garbage Collection: http://en.wikipedia.org/wiki/Garbage_collection_(computer_science)

〔5〕 M.I.Cohen, "Advanced carving techniques", Digital Investigation Volume 4, Issues 3-4, pp 119-128, August 2007.

〔6〕 M.I.Cohen, "Advanced Jpeg Carving", e-Forensics08: Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop, pp. 1-6 ICST(Institute for Computer Sciences Social-Informatics and Telecommunications Engineering), January 2008. ISBN 978-963-9799-19-6.

〈著者紹介〉

김 상 대 (Sang-dae Kim)
2006년 2월: 명지대학교 소프트웨어 공학과 졸업
2009년 3월~현재: 고려대학교 정보경영공학전문대학원 석사과정
〈관심분야〉 디지털 포렌식, 모바일 포렌식

변 근 덕 (Keun-duck Byun)
2004년 2월: 아주대학교 학사 졸업
2006년 2월: 고려대학교 석사 졸업
2006년 3월~현재: 고려대학교 정보경영공학전문대학원 박사과정
〈관심분야〉 임베디드 포렌식, 역공학

이 상 진 (Sang-jin Lee)
1987년 2월: 고려대학교 수학과 졸업
1989년 2월: 고려대학교 수학과 이학석사
1994년 8월: 고려대학교 수학과 이학박사
1989년 10월~1999년 2월: ETRI 선임 연구원
1999년 3월~2001년 8월: 고려대학교 자연과학대학 조교수, 부교수
2001년 9월~2006년 8월: 고려대학교 정보보호대학원 부교수, 교수
2006년 8월~현재: 고려대학교 정보경영공학전문대학원 교수
2008년 3월~현재: 고려대학교 정보보호연구원 디지털포렌식연구센터장
2006년 1월~현재: 한국디지털포렌식학회 편집이사
〈관심분야〉 디지털 포렌식, 모바일 포렌식, 심층 암호, 해쉬 함수