Research In Motion's BlackBerry Wireless E-Mail Devices: Decoding the Mobitex protocol

Kingpin, @stake, January 2002

1 Introduction

The Research In Motion BlackBerry wireless e-mail device product line is split into two versions:

- **Enterprise Edition** is designed for corporate environments using Microsoft Exchange or Lotus Domino. Marketing literature states that Triple DES is implemented to provide end-to-end encryption of the e-mail message between the mail server and the BlackBerry device.
- **Internet Edition** is sold through select service providers (e.g., Cingular Interactive, Compaq, and EarthLink) and is bundled together with an Internet/Web-based email account. All mail passes through the ISP, which is then forwarded to the correct location.

Our initial research of the BlackBerry device focuses on the wireless transmissions between the device and the Cingular Wireless Network (the wireless backbone for BlackBerry, here after referred to as the "base station")[1]. This document serves as an introduction into the BlackBerry device and the configuration needed to decode the wireless protocol. It does not cover any of the back-end server configuration or middleware running on the device itself.

The bulk of this research is based on "The Inherent Insecurity of Data Over Mobitex Wireless Packet Data Networks" paper released anonymously in 1997 to the *rec.radio.scanner* Usenet newsgroup and available on the web at `http://atomicfrog.com/archives/exploits/rf/MOBITEX.TXT`. The paper contains detailed technical information on the Mobitex protocol, radio frequencies, computer interfacing, and source code data decoding. It should be read in parallel with this document.

The goal of this research was primarily to look for potential security risks. It was successful. The Internet Edition devices transmit their information in the clear, which was discovered and verified by sending messages from our own device and monitoring the radio transmissions going to the base station. It is also possible to decode e-mail attachments, which are also sent in the clear. However, it appears that the Enterprise Edition devices do use some sort of encryption around the message body, though analysis of this portion has not taken place. Message header information is always sent in the clear, regardless of if encryption is used around the body.

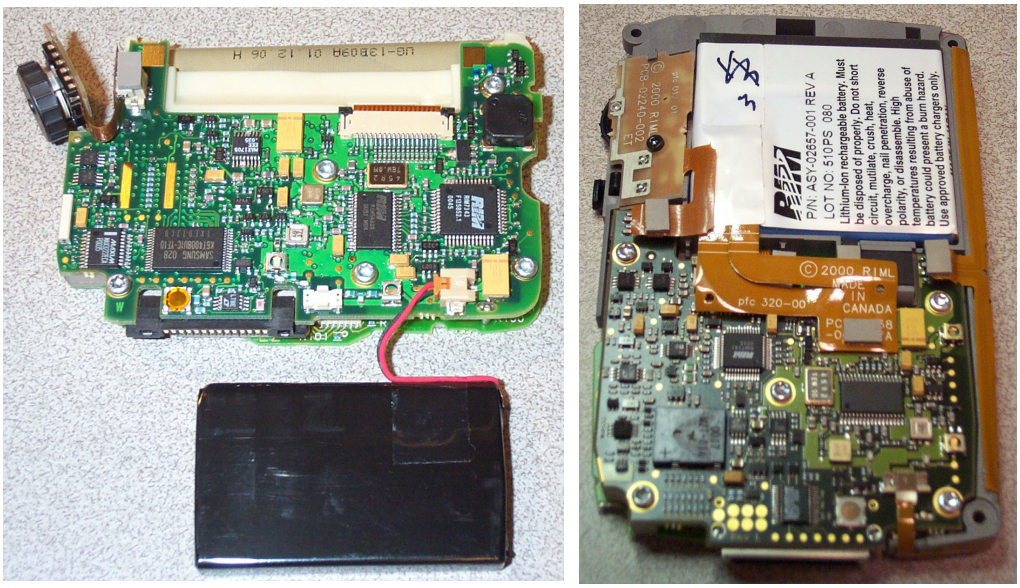
2 Device Details

As of this writing, two BlackBerry devices exist: the RIM 950 and RIM 957 (Figure 1). They are both based around the Intel 386 architecture. The RIM 950 has 4MB of ROM while the RIM 957 has 5 MB. Both have 512kB of SRAM. The RIM 950 is a smaller pager-sized device with a 132 x 65 pixel LCD screen (8 line x 25 characters) and powered by a single AA battery. The RIM 957 is more of a PDA form-factor, containing a larger 160 x 160 pixel LCD screen and internal rechargeable Lithium battery. Each also has a track-wheel for scrolling/selecting, miniature QWERTY-style keyboard, and an RS232 serial port for application/ OS loading and device configuration. Other than physical characteristics, the functionality of both devices is essentially identical.

---

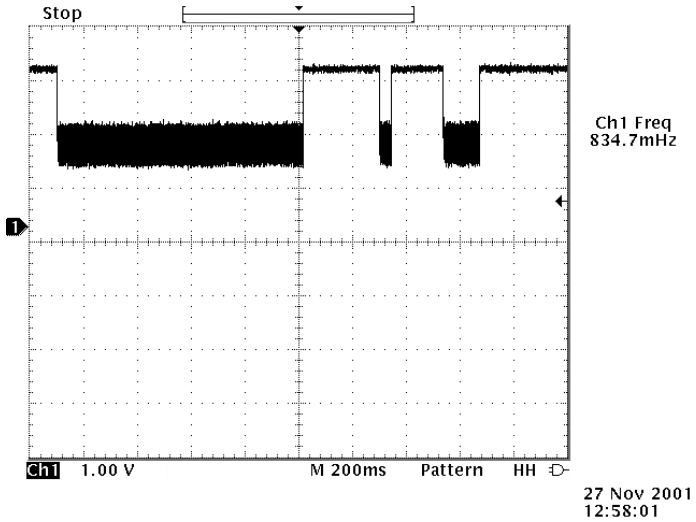
[1] This network was formerly called the BellSouth Intelligent Wireless Network

Figure 1: Circuit internals of RIM 950 (left) and RIM 957 (right)



Both units also have a set of 7 test points on the printed circuit boards that appear to be used for programming or testing. Initial research of these points yielded signals during device transmit and reception (Figure 2). However, since our research was focused mainly on the wireless transmissions component, only initial probing took place. These test points are easily accessible through the back panels of the device housing. The 950 requires some plastic to be removed, but the 957 already has holes drilled into the housing and is simply covered by a piece of tape and external sticker.

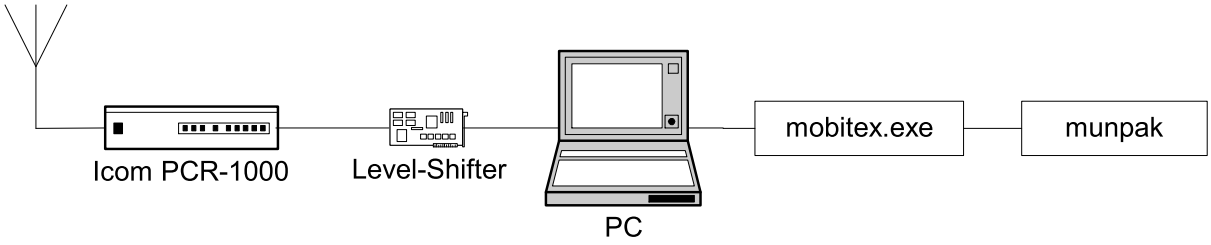
Figure 2: Oscilloscope screenshot showing data on test point as device is transmitting



3 Data Decoding

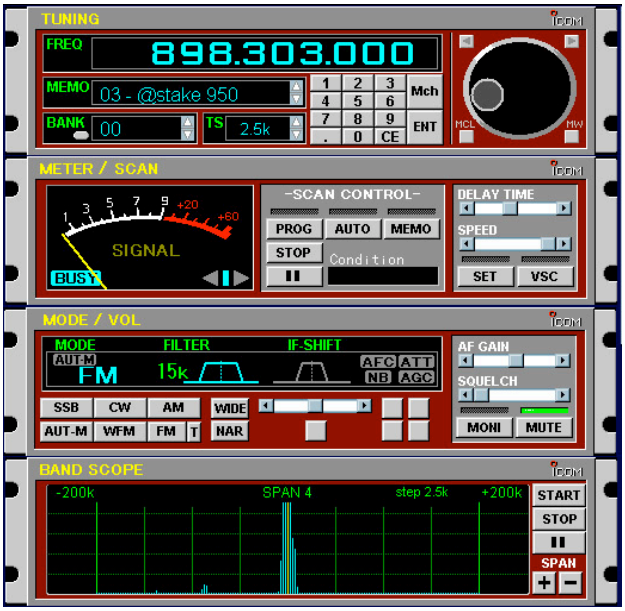
Figure 3 shows our setup used to receive and decode Mobitex/BlackBerry transmissions.

Figure 3: Experimental setup for data decoding



The Icom PCR-1000 is a software-controlled (Figure 4) wideband radio receiver (100 kHz-1.3 GHz). Any number of scanners/radio receivers can be used, but the PCR-1000 has an unfiltered audio output, which provides a clean signal for use in data decoding applications. Unfiltered audio is necessary for the decoding of data sent at high rates, such as the 8000bps Mobitex protocol. Many radios can be modified to provide a similar undistorted signal, often referred to as a “Discriminator Output”.

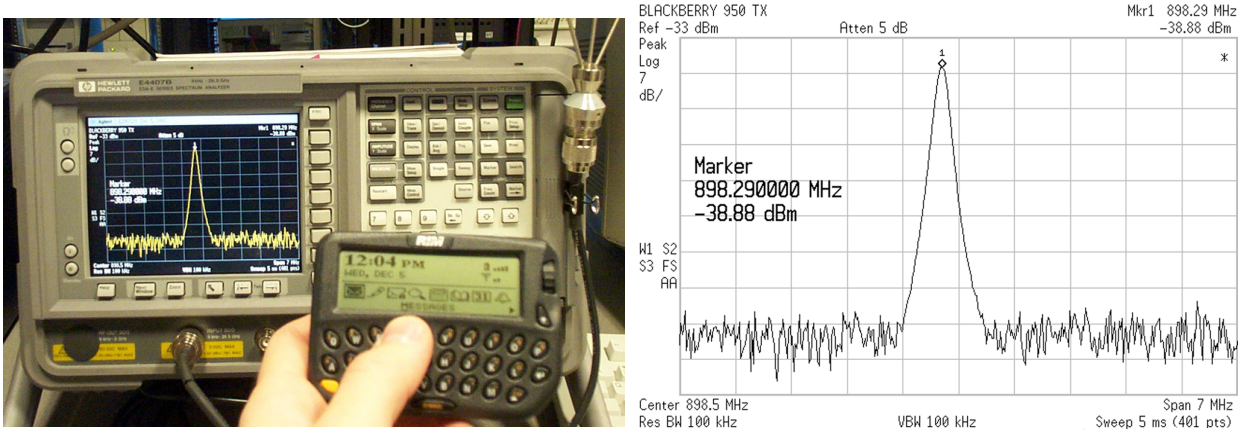
Figure 4: Screenshot of the PC-based PCR-1000 control software



The BlackBerry devices are designed to operate on the 900 MHz Mobitex networks. The transmit (TX) range is 896-902 MHz, 12.5 kHz channel spacing, and the receive (RX) range is 935-941 MHz, 12.5 kHz channel spacing. To decode transmissions coming from the base station, we would simply monitor the frequency range of 935-941 MHz. To monitor the transmissions coming directly from a particular BlackBerry device requires an additional step. Although the TX frequency will change to the strongest available signal as directed by the wireless network (according to the Mobitex specification), we can narrow down the number of possible frequencies used in our area and monitor them all when it is time to decode the data being transmitted from the device.

A spectrum analyzer is used to graphically display the signal power of an RF transmission over a frequency domain. In our research, the spectrum analyzer was used to measure and determine the current TX frequency of the BlackBerry device (Figure 5). The spectrum analyzer shows the spectral peak of transmission being at 898.29 MHz at the given point in time. This peak appears when the device is transmitting, and will disappear when the device stops.

Figure 5: Spectrum analyzer capturing BlackBerry signal (left) and screenshot of TX center frequency (right)



Simple circuitry is needed to convert the audio signal from the radio receiver into the proper levels for computer interfacing. This level-shifter hardware, also known as a “POCSAG Decoder” or “Hamcomm Interface”, is powered by and connects to a standard PC serial port. Figure 6 shows the schematic for the circuitry, which can be built with a few dollars worth of common components available at any electronics store. Figure 7 shows the assembled unit used for our research (which was built using the POCSAG Decoder PCB sold by L0pht Heavy Industries from 1996 to 1997).

Figure 6: POCSAG Decoder/Level-Shifter Schematic

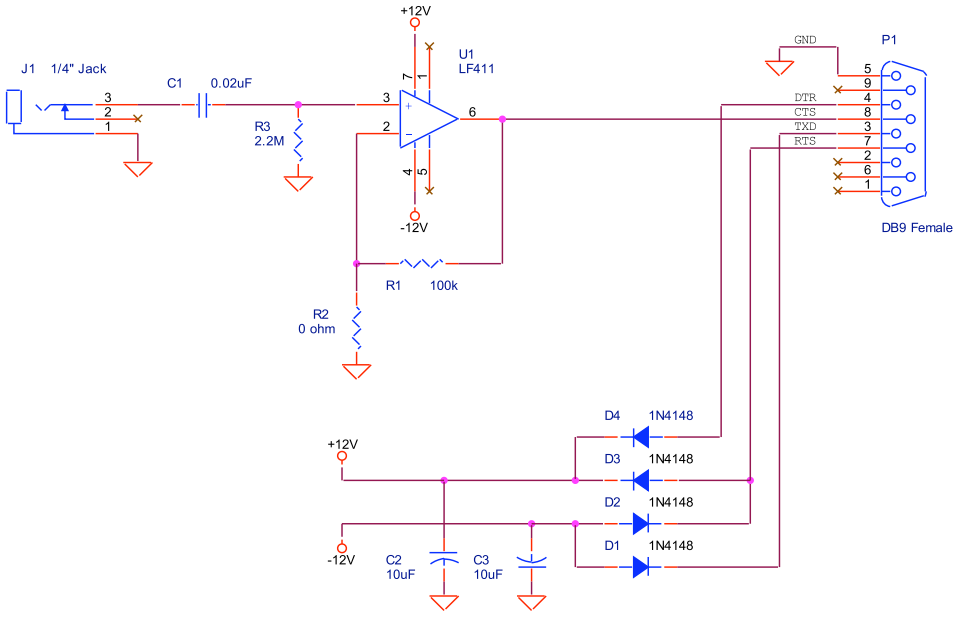
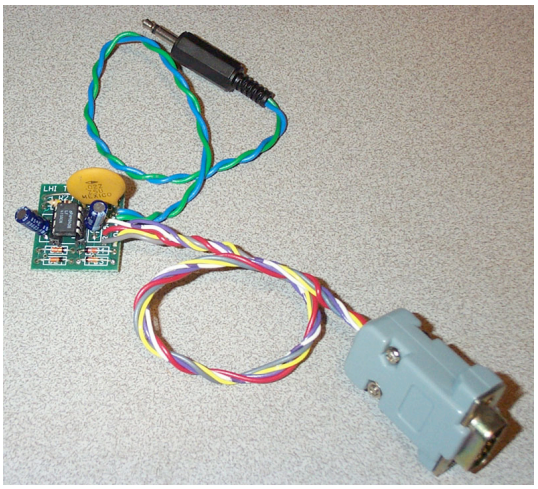


Figure 7: Level-Shifter Assembled Circuit



The actual data decoding software, `mobitex.exe`, is run on the PC that the level-shifter circuitry is connected to. This software, released with the anonymous 1997 paper mentioned earlier, will take the raw audio signal from the scanner radio (conditioned by the level-shifter) and decode the data as defined in the Mobitex protocol specification. The output from the `mobitex.exe` tool is an ASCII hex dump of the Mobitex data packet (MPAK). All the higher-level Mobitex protocol information is stripped out, leaving just the raw data information that has been transmitted. Our test setup using a Pentium III 450MHz was sufficient for accurate decoding. A 386SX 25MHz was not.

Due to the characteristics of Mobitex, different command line settings are required when decoding transmissions from the base station as opposed to directly from the BlackBerry device. The `/SY:3333` option should be used when mobile transmissions are being monitored.

4 Mobitex Data Packet (MPAK) Parsing

A post-processing step is required in order to parse the data output by `mobitex.exe`. The tool released with our paper, entitled `munpak`, separates the MPAK into its individual components and displays all known information in a useful, readable format. The tool was written to make it easier to understand and visualize what data components are being transmitted in the clear and what components are not. The details of the MPAK structure are available in the `munpak` source code. Below are the usage instructions for the tool:

```
C:\>munpak -h

@stake [kingpin@atstake.com], January 2002 v1.00

NAME
        munpak - A post-processing tool for parsing Mobitex data
        packets (MPAKs)

USAGE
        munpak [ options... ] <raw mobitex filename>

OPTIONS
        -h              : you're reading it
        -m man          : specific man to parse [default: parses all]
```

The input file to `munpak` is an unmodified version of the `mobitex.exe` output, which looks like:

```
FD236881B808FD23680186BF00020000002510DF        ý#h¸.ý#h.†¿.....%.ß
00000000020002202007473130313130 3100A357       ....... .G101101.£W
07AFFFAB5005434D494D4503408080805400A303        .¯ÿ«P.CMIME.@€€€T.£.
000010C0004C021004136B696E6770696E405D94        ...À.L....kingpin@]”
61747374616B652E636F6D01093136353839612C        atstake.com..16589a,
3637320007043C1116E40803466F6F0B010151BA        672...<..ä..Foo...Q°
F1044B8317940001020201000F53656C6C20A06B        ñ.Kƒ.”.......Sell  k
746865206661726D2E0A1000000000000000DE5E        the farm.........Þ^
```

The output of `munpak` is in an easily readable form as follows:

```
================================================================================

Radio Oriented Synchronous Information (ROSI) Header
----------------------------------------------------
Mobitex Access Number (MAN): 16589672
Frame ID: 129
Sequence Number: 184
Data Blocks: 8

Mobitex Packet (MPAK) Header
----------------------------
Sender MAN: 16589672
Addressee MAN: 100031
Flags: None
Traffic State: N/A
Packet Type: Data
Time Stamp: N/A
Packet ID: 37

Mobitex Packet (MPAK) Body
--------------------------
Destination MAN: G101101
Message Type: E-Mail Original (MIME)
To: kingpin@atstake.com
From: 16589672
Subject: Foo
Body: Sell the farm.

================================================================================
```

With the particular example given above, this data was captured directly from our RIM 950 Internet Edition device. The e-mail was sent to `kingpin@atstake.com` from Mobitex Access Number (MAN, a unique identifier) `16589672`. All e-mail information, including the `To:`, `From:`, `Subject:`, and `Body:` fields are transmitted in the clear without any encryption or obfuscation. A lot of useful and interesting information can be seen when the MPAK is decoded. The Radio Oriented Synchronous Information (ROSI) header and MPAK header are always sent in the clear. If encryption is used, only the MPAK body is encrypted.

The `-m` option can be used with `munpak` to filter and decode the MPAKs only intended for the specified MAN. All other data will be ignored. Additionally, the `munpak` source code was designed to be modular to allow new message types and data structures to be added if they are encountered at a later date.

5 Summary

Our research has demonstrated that certain versions of the BlackBerry device transmit their e-mail messages in the clear. We are using common off-the-shelf equipment and a few pieces of data decoding software. Additionally, it appears that the Enterprise Edition devices use some sort of encryption around the message body *only* (leaving the ROSI and MPAK headers in plaintext).

All available Research In Motion literature mentions end-to-end encryption functionality of the BlackBerry Enterprise Edition. In no literature is it mentioned that the Internet Edition uses encryption, but it is *not* mentioned that it doesn't. The lack of clarification regarding the Internet Edition may be leading users to assume their data is indeed encrypted, which clearly is not the case.

A Resources

1 Research In Motion, `www.rim.net`

2 BlackBerry Cryptographic Kernel Policies, `http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp312.pdf`

3 The RIM BlackBerry Serial Protocol, `www.off.net/cassis/protocol-description.html`

4 M. M. Khan's Wireless Data Over RAM's Mobitex Network paper, `http://spie.org/x648.html?product_id=228159`

5 Mobitex Operators Association, `www.mobitex.org`