Security vulnerabilities and security solutions for medical pagers

Sam D. Kim[1], **Tyler Bletsch**[2]

[1] Trinity College of Arts & Sciences, Duke University, Durham, NC, USA
[2] Department of Electrical & Computer Engineering, Pratt School of Engineering, Duke University, Durham, NC, USA

Duke | ELECTRICAL & COMPUTER ENGINEERING

Pagers receive sensitive patient data, yet lack even most basic security

Though largely replaced by smartphones and email, one-way pagers remain very common in niche fields—like hospitals, where they are favored for their low battery use and reliable connectivity amid radiation-shielded walls.

Despite receiving protected health information (PHI) in real time, most pagers lack even the most basic security, and most sensitive data are transmitted unencrypted.

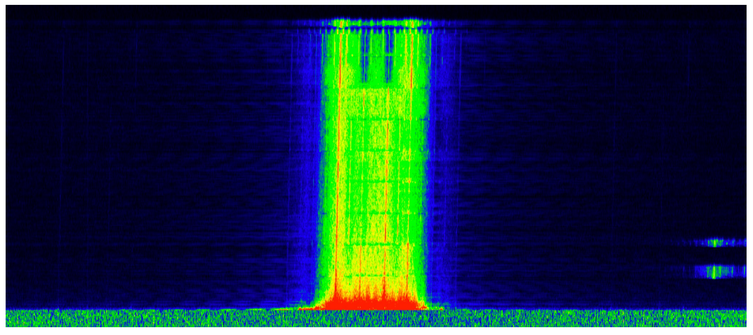
We illustrate this **dangerous yet neglected security flaw** by:

- Demonstrating the ease of intercepting and decoding pages
- Building a simple proof of concept for pager security

Background

Most U.S. pagers use either **FLEX** or **POCSAG** as their paging protocol. We can identify pager frequencies by their characteristic sounds and waterfall shapes, which vary by protocol.

We used the FLEX frequency **929.577 MHz** (waterfall below), which appeared to be used by a paging network supporting hospitals in North and South Carolina. The frequency had relatively busy traffic, with roughly about 50 pages a minute.

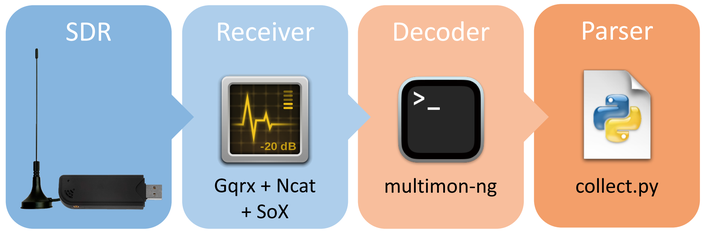


Intercepting patient data with just $15 and a computer

We intercepted, decoded and parsed pages using a USB-based **software-defined radio (SDR)**—available for about $15 on Amazon—and some free Linux software, as follows:

- Tune SDR to desired frequency with Gqrx, and stream the audio over UDP
- Capture and resample audio with Ncat and SoX
- Decode data with multimon-ng
- Parse multi-part messages with Python script

A more detailed protocol, as well as all scripts used, are available on GitHub: https://github.com/sam-k/pager-sec




```
[samkim@Sams-MBP:SDR$ cat flex_collected.txt
FLEX|3200/4|09.104.A|[redacted]|LS|5|ALN|3.0.K
└[redacted]

FLEX|3200/4|09.104.A|[redacted]|LS|5|ALN|3.0.K
└[redacted]

FLEX|3200/4|09.104.A|[redacted]|LS|5|ALN|3.1.F
FLEX|3200/4|09.105.A|[redacted]|LS|5|ALN|0.0.C
└[redacted]
```

The decoded messages revealed, among others:

- Patient names, room numbers, other identifying info
- Vital signs, lab results, diagnoses, drug prescriptions
- Names of physicians, nurses, hospitals

```
FLEX|3200/4|08.103.C|0004783821|
LS|5|ALN|3.0.K|PT IN 413 DOE,
JANE 37F DILAUDID 0.5MG 1HR AGO
STILL C/O PAIN, INCREASE DOSE?
```

*Typical, but not real, message.

Security proof of concept

We built a proof of concept for encrypting and authenticating pager communications, which are limited by the following:

- Small memory, which may make computationally or resource-intensive protocols infeasible
- One-way communication, which precludes many common encryption protocols (e.g., Diffie–Hellman key exchange)

To show even limited hardware can support authenticated encryption, we modeled pagers with **Arduino Nano**, which has only 32 KB of flash memory and 2 KB of SRAM. (Compare with the Motorola Advisor pager, which has 8×32 KB of SRAM.)

We used **ChaCha20–Poly1305**, a lightweight symmetric-key protocol for authenticated encryption. This requires paging system and each pager to know a unique pre-shared key.

Actual pagers today have much more memory than an Arduino Nano, which may allow more expensive, asymmetric-key protocols such as RSA (with pre-shared keys).

Easy to Attack

With cheap hardware and free software, anyone can intercept pages to access sensitive data.

Easy to Defend

Even limited hardware can support authenticated encryption for one-way communication.