Why (special agent) Johnny (still) Can't Encrypt: A security analysis of P25

Matt Blaze, with Sandy Clark,

Travis Goodspeed, Perry Metzger,

Zach Wasserman and Kevin Xu

University of Pennsylvania

Contact: **blaze@cis.upenn.edu**

Security in Open Telecommunications Networks

- Research project at University of Pennsylvania
  - Part of joint project with Penn State University
- Aim is to analyze and improve security in various wireless networks (cellular, two-way, etc)
  - U. of Pennsylvania's focus is on two-way public safety radio
- Funded by National Science Foundation
  - CNS-0905434
- Unclassified project
  - Relevant findings will be published in open literature

APCO Project 25 ("P25")

- Standard (in the US and elsewhere) for digital two-way radio (voice and low-speed text)
  - Widely fielded by gov't: local police & fire, Federal law enforcement & security services, DoD.
  - Standards under ongoing development since early 90's
  - P25 products increasingly available since early 2000's
- Drop-in replacement for analog FM systems
  - Uses narrow band channels, limited infrastructure
  - Can use simplex, repeaters, or trunked infrastructure
- Cryptographic security options
  - Content confidentiality (encryption)

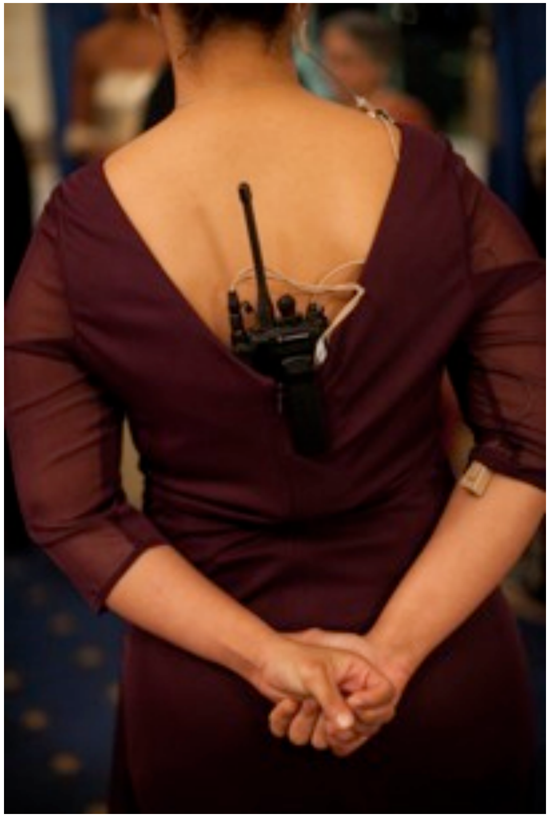
P25 Equipment

- Wide range of COTS subscriber radios available
  - Mobile, portable, base and infrastructure
- Several US vendors; Motorola dominates in Federal law enforcement sector
- Equipment features, user interfaces (somewhat) standardized across vendors

Typical handheld P25 radio:
Motorola XTS-5000

P25 Users: Not Just Local Agencies

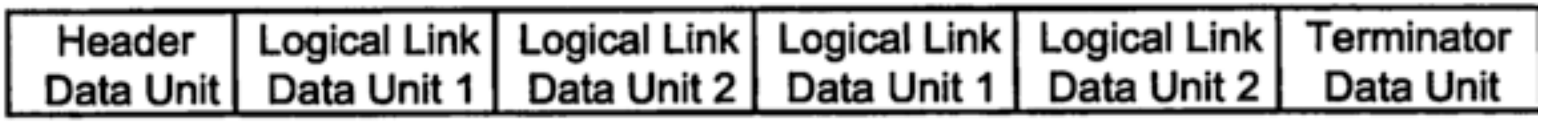
DoD: Warfighters, Security
(photo: Lindsey Addario, NY Times)



DoJ, DHS: LE Surveillance, Exec. Protection, etc.
(Photo: Pete Souza, The White House)

The P25 Voice Protocol

- Narrow-band radio channel (12.5Khz)
  - co-exists with analog FM
  - 9600 bps (4800 2 bit symbols/sec)
- IMBE vocoder
  - reasonable quality speech
  - train of 1728 bit voice frames that encode 180ms of audio



P25 Security

- Symmetric encryption
  - Unclassified: AES, DES, etc.
  - Classified: various Type I
- Traffic keys must be loaded into radios in advance
  - Via keyloader device or over-the-air rekeying (OTAR)
  - Keys can expire, self destruct
- No "sessions"
  - Sender radio selects crypto mode & key
  - Up to receiver to decrypt
- Received cleartext always demodulated & played
- Received ciphertext decrypted & played if correct key available
- No authentication
- Sender's radio makes **all** security decisions
  - Radios can be configured for always clear, always encrypted, or user-selected
  - User-selected is std config

P25 Security

- Our paper examines in detail
  - Apparently ad hoc design
    - – no formal (or informal) security requirements in std
  - Traffic encryption *itself* isn't obviously broken
- But does suffer significant protocol weaknesses
  - – No authentication
  - – Susceptible to traffic analysis
    - radio unit IDs sent in clear even when encryption enabled
  - – Vulnerable to very efficient denial of service
    - 14dB energy advantage to **attacker**
- Serious crypto-usability weaknesses

Some practical attacks

No Authentication

- Voice traffic can be encrypted for confidentiality, but is not authenticated
  - No assurance that it came from authorized user
  - No protection against replay, splicing, etc.
- Displayed received Unit ID not authenticated
- Inbound clear traffic on channel *always* accepted, even when radio is in secure mode
- AES-GCM crypto mode doesn't fix this.

Passive and Active Traffic Analysis

- Subscriber radio's Unit ID, TalkGroup ID, NAC sent with every transmission
  - 24 bit unit ID is typically unique to each radio
  - Effectively identifies individual radio + agency it belongs to
- Standard supports encryption of Unit ID
  - But we found UID *always* in clear, even if crypto enabled
- Radios typically automatically respond to pings
  - Active adversary can easily discover idle radios
  - Transparent to pinged radio

Scenario: "Maurauder's Map"

- Ping response is sufficient to allow automated direction finding of targeted radios
  - requires two bases at fixed locations with phased directional antenna
- Adversary can thus create a real-time map of selected radios, even when they are "idle"
- Significant potential threat in military environment

Denial of Service (in theory)

- P25 uses aggressive error correction codes
  - But individual subfields of transmission are error-corrected *separately*
- Adversary can select a single subfield to jam within frame
  - Pattern at start of transmission makes synchronization easy
- Voice frame is 1728 bits, including critical 64 bits *NID* subfield that IDs frame type
  - Jamming 64 bits is renders *entire* 1728 bit frame unreadable
- That's just 32 symbols of jamming per 864 symbols
- Jammer needs 14dB **less** energy than the transmitter
  - Compare: Analog FM requires (about) **equal** energy to jam
  - Jamming digital spread spectrum requires **much more** energy.

Denial of Service (in practice)

- How hard is it to build a P25 subframe jammer?
- TI CC1110 is a single-chip digital radio transceiver chip
  - supports native protocol very similar to P25
  - sufficiently close to recognize start of P25 frames…
- Used in GirlTech IMME toy instant messenger ($15)
  - So we developed our own P25 jammer firmware…
  - “My First Jammer”

Scenario: Selective Jamming

- Need not jam every P25 transmission
- Jammer is low duty cycle
  - spends most time in receive mode
  - can be programmed to recognize certain types of transmissions and interfere only with them
- Easy to configure a jammer that recognizes and disables only encrypted P25 signals
  - force users to switch to clear in order for communication to work

Usability in Practice

Ø

Potential usability problems

- Poor feedback about crypto state
  - *Transmit* crypto is controlled by an obscurely marked toggle switch
  - Switch's state has no effect on *received* audio
    - clear signals always accepted in encrypted mode
    - encrypted signals accepted in clear mode (if keyed)
- Frequent rekeying + unreliable rekeying
  - many agencies use short-lived keys
  - but re-keying is difficult and unreliable

Poor Crypto Feedback (see Whitten & Tygar, '99)

- Current radios are typically configured to control outbound crypto with a two-position switch
  - Often obscurely marked, out of view
- Little feedback to user about crypto state other than the switch itself
  - "Encrypted" icon on display
  - Configurable "clear" beep warning
    - but same beep is also used to indicate other things
- Little chance for other users to notice or help
  - Received cleartext always accepted, even when their own switch is in the "secure" position

Example: Motorola XTS5000:

Crypto switch

Display

Example: Motorola XTS5000:

Crypto switch

Display

Cumbersome Keying & Keying Failure

- Groups frequently attempt to use encrypted mode, but discover they cannot because one or more team members' radios does not have current key
- P25 radios cannot be hand rekeyed by user in field
  - Traffic keys must be loaded by KVL or using OTAR protocol
  - OTAR frequently fails
- Rigorously enforced key expiration and key replacement policies actually make it *more* likely that some users will not have current key material
  - Some agencies perform monthly or even weekly rekeys

No Ad Hoc Field Keying

- If even a single user lacks current keys, there is usually nothing a team can do
  - Keys *cannot* be created or entered by hand into radio
  - Keyloader hardware is not typically available in field.
  - OTAR frequently fails in practice
- Often only practical option is for an entire operation to go to clear

Motorola KVL-3000
P25 Keyloader

“Rule #1 of cryptanalysis:
First, look for cleartext”

Bob Morris, NSA
(invited talk at Crypto ’95)

P25 COMSEC in Practice

- The P25 traffic analysis and efficient DoS attacks we found are potentially serious, but require some expertise and resources on part of adversary
  - **Current** off-the-shelf equipment can't easily implement most of the protocol-level attacks we found without modification
    - inexpensive software-defined radio will soon change this, however
  - Not much can be done to mitigate these vulnerabilities without changing the P25 protocols in any case
- More serious are usability weaknesses that can be easily exploited by anyone, *today:*

A significant volume of law-enforcement-sensitive cleartext regularly goes over the air, with users unaware

Unintended Sensitive P25 Cleartext

- Last year, we accidentally misconfigured a P25 radio in our lab, and were surprised to hear chatter from a federal tactical surveillance operation.
  - This turned out not to have been a fluke event.
- We subsequently collected statistics about unintended over-the-air sensitive cleartext in several metropolitan areas
  - Focused on confidential tactical law-enforcement traffic
    - omitted local agencies, non-covert operations (e.g, interop networks, uniformed FPS patrols), etc.
  - No encrypted traffic captured
  - Used only readily-available, unmodified consumer-grade equipment
  - Live monitored samples of traffic, recorded traffic statistics

Intercepting the Federal Spectrum

- 2000 discrete VHF and UHF voice channels allocated to Federal government
  - 24 MHz of spectrum
  - 12.5 KHz channels
  - Law enforcement mixed in among less sensitive users
    - Some agency channels are widely known, others not
- Easy to identify the channels used locally for covert tactical LE activities

- Many P25 receivers on market
- Icom R-2500
  - Aimed at hobby “scanner” market, includes P25 option
  - Legally available to anyone

Results

- We searched the Federal VHF and UHF spectrum for the frequencies used for sensitive tactical networks
  - Likely candidate frequencies easy to identify: they carry mostly encrypted traffic
- Configured a small network of R-2500 receivers in several metropolitan areas with software to systematically scan these networks and log incidence of cleartext
  - Periodically "live monitored" samples of cleartext audio
  - Did not retain identifiable information about agents or targets
- In each metropolitan area:
  - Most tactical traffic was apparently successfully encrypted
  - But still > 30 minutes (mean) sensitive cleartext per city per day
    - high variance; lower volume on weekends and holidays.

How Sensitive is Sensitive?

- The P25 unintended cleartext we live-sampled included some of the most sensitive investigative data the gov't handles:
  - Names and/or identifying features of targets and confidential informants, their locations, descriptions of undercover agents
  - Information relayed by Title III wiretap plants
  - Plans for forthcoming takedowns and operations
  - Wide range of crimes, some involving targets that appeared to employ reasonably sophisticated countermeasures
  - Sensitive cleartext captured from virtually *every* DoJ & DHS LE agency
- Mostly criminal SBU, but some national traffic possible:
  - Executive protective details
  - Some counter-terror, counter-intelligence targets

What is Going Wrong?

- Three categories of unintended cleartext:
  - **Single user error**: one user transmitting in clear, but communicating with an encrypted team
  - **Group error**: everyone in clear, indicated they were encrypted, no one noticed they weren't
  - **Keying failure**: one member of group did not have key, so everyone went to clear
- Cleartext we sampled was roughly evenly split between single/group error and keying failure

Observations

- P25 tactical radio crypto capability is now widely deployed by federal law enforcement
- Yet Federal P25 networks still carry quite a bit of easily intercepted LE sensitive cleartext
- Two dominant causes, each requiring different mitigation approaches
  - Accidental cleartext (about half the time)
  - Keying failure (about half the time)

Mitigation Strategies

- Going forward, P25 protocols & products require a top-to-bottom redesign for security
  - Until then, P25 systems should not be considered reliably secure; end users should understand this
- In the short term, some adjustments to keying practices and radio configuration could significantly reduce incidence of unintended sensitive cleartext
- **http://www.crypto.com/p25/**

Specific Recommendations (1) Accidental Cleartext

- Usability problems might be partly mitigated with improved training, user awareness
  - But fault is ultimately with the radios, not users
- Configure radios to simplify the crypto UI
  - Disable the separate "secure" switch
  - Instead, configure radios to "strap" crypto on/off
    - always-on or always-off on a per-channel basis
    - label channel names accordingly (e.g., "TAC1 Secure")

Specific Recommendations (2) Keying Failure

- Centrally-controlled keying is often not compatible with practical need for flexible tactical options
  - Make keyloaders available in field to agents & teams
- *Decrease* frequency of mandatory rekeys via OTAR
  - Current practice with expiring keys has demonstrably *reduced* security by making it less likely that every authorized user in a group has current key material.
  - Instead, rekey only when needed (e.g., lost radio).
  - In general, deploy long-lived, non-volatile keys
    - increases chance that users who need to communicate securely will already share a common key when they need it.

In the longer term:
these problems exist because radio encryption
is harder than we think

What's different about P25?

- Observe that although it's used for "two-way" radios, P25 is really a "one-way" protocol
  - sender unilaterally picks all security parameters (or not) and broadcasts away
  - usability: receiver might not notice if crypto is off
- We don't know much about designing one-way protocols
  - almost all of crypto community's design wisdom is for two-way protocols

Typical Crypto Protocol Properties

- Cast of characters: Alice, Bob, Eve, etc
  - Alice and Bob have roughly equal power
- Bilateral session negotiation
  - both parties contribute, either can halt things
- Can conservatively start with most secure configuration & negotiate *downward* from there
  - if result is unacceptable to either side, we stop
- Frequent roundtrips during session

But, one-way protocols can do none of this

Even worse: One-way protocols preclude conservative designs

- Two-way protocol: try most secure possible configuration first
  - if other side can't do it, all is not lost
  - negotiate to mutual satisfaction
- One-way protocol: sender must assume no more security (or keys) than receiver is likely to have
  - if primary goal is reliable communication, sender's best choice is always to transmit in the clear

Can we get the benefits of negotiated "sessions" in the one-way context?

Error detection/recovery matters a lot

- Once the cleartext is sent, the damage may already be done
- Receiver is in good position to detect failure
  - but is not part of the protocol

Can we create protocols that are one-way in normal operation, but allow intervention in case of error?

We've been here before (and the news isn't good)

- What are other one-way protocols?
  - email encryption
    - not exactly our finest hour
  - direct broadcast video
    - always encrypted; receiver's problem if it doesn't work
    - hard to apply to other protocols

The one-way model seems an important and largely ignored corner of the protocol design space

Contact

Matt Blaze

University of Pennsylvania

3330 Walnut St

Philadelphia, PA 19104

blaze@cis.upenn.edu