Authloop:

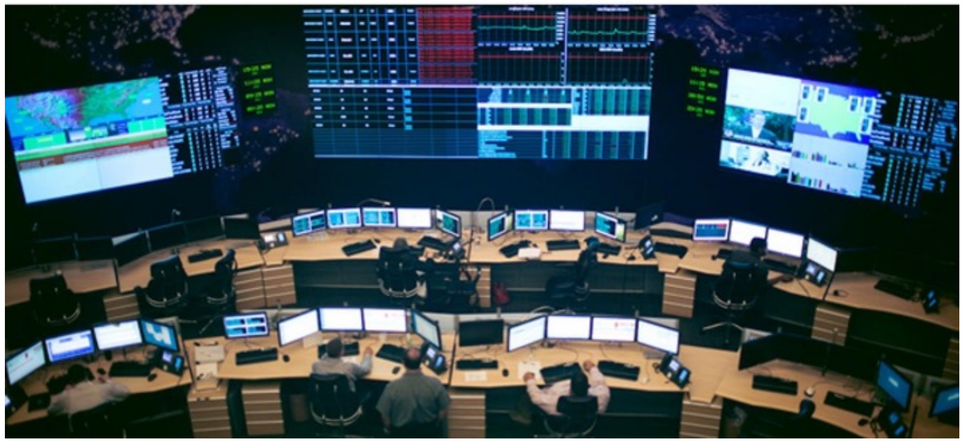
Practical End-to-End Cryptographic Authentication for Telephony over Voice Channels

Brad Reaves, *Logan Blue, Patrick Traynor*

Phones are really trusted…

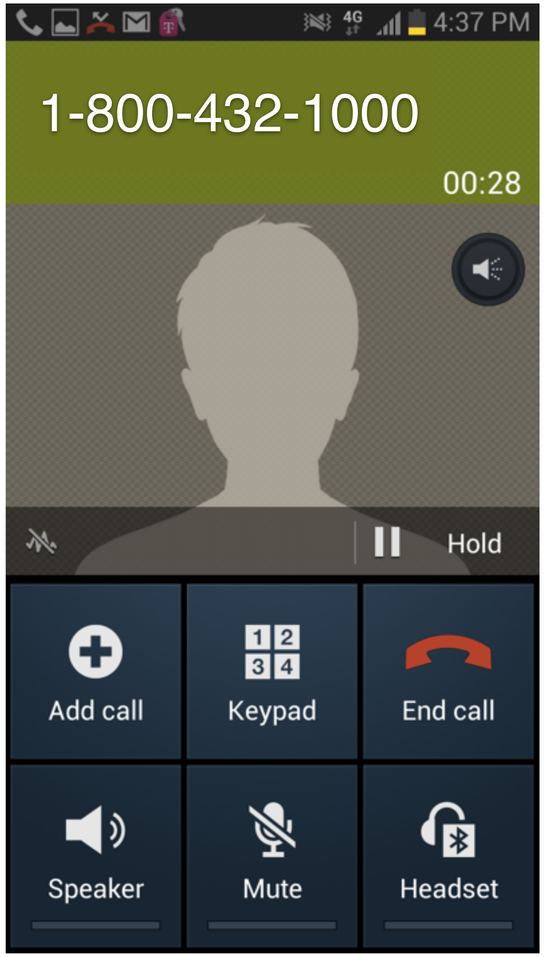
- Phones are our backup, our trusted platform…
  - Customer service — Account Reset
  - Network operators
  - Critical Infrastructure

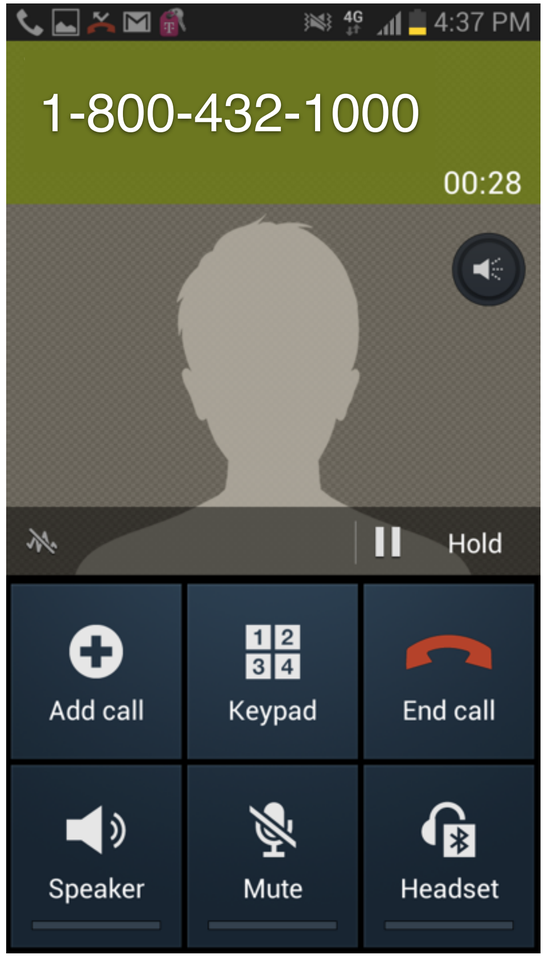






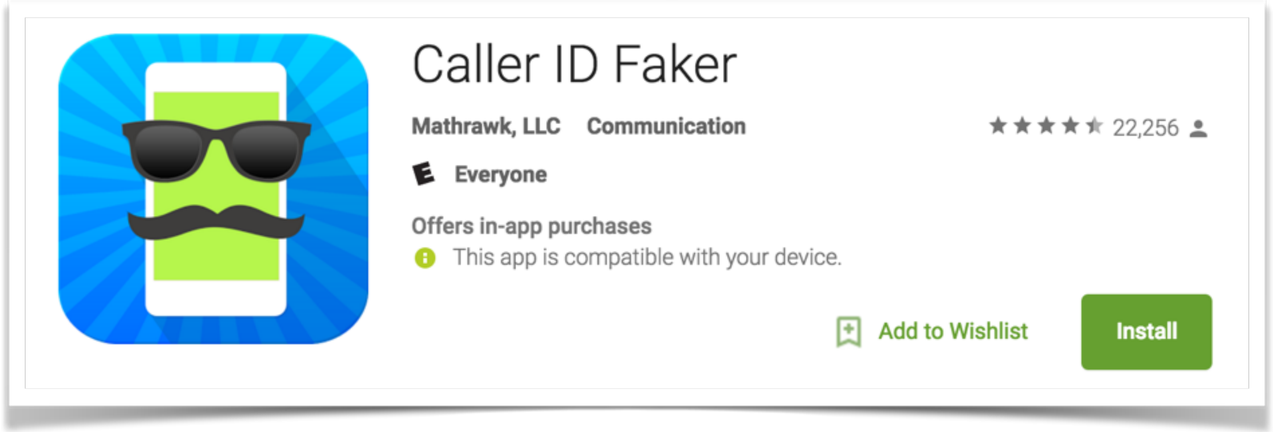
But they shouldn't be





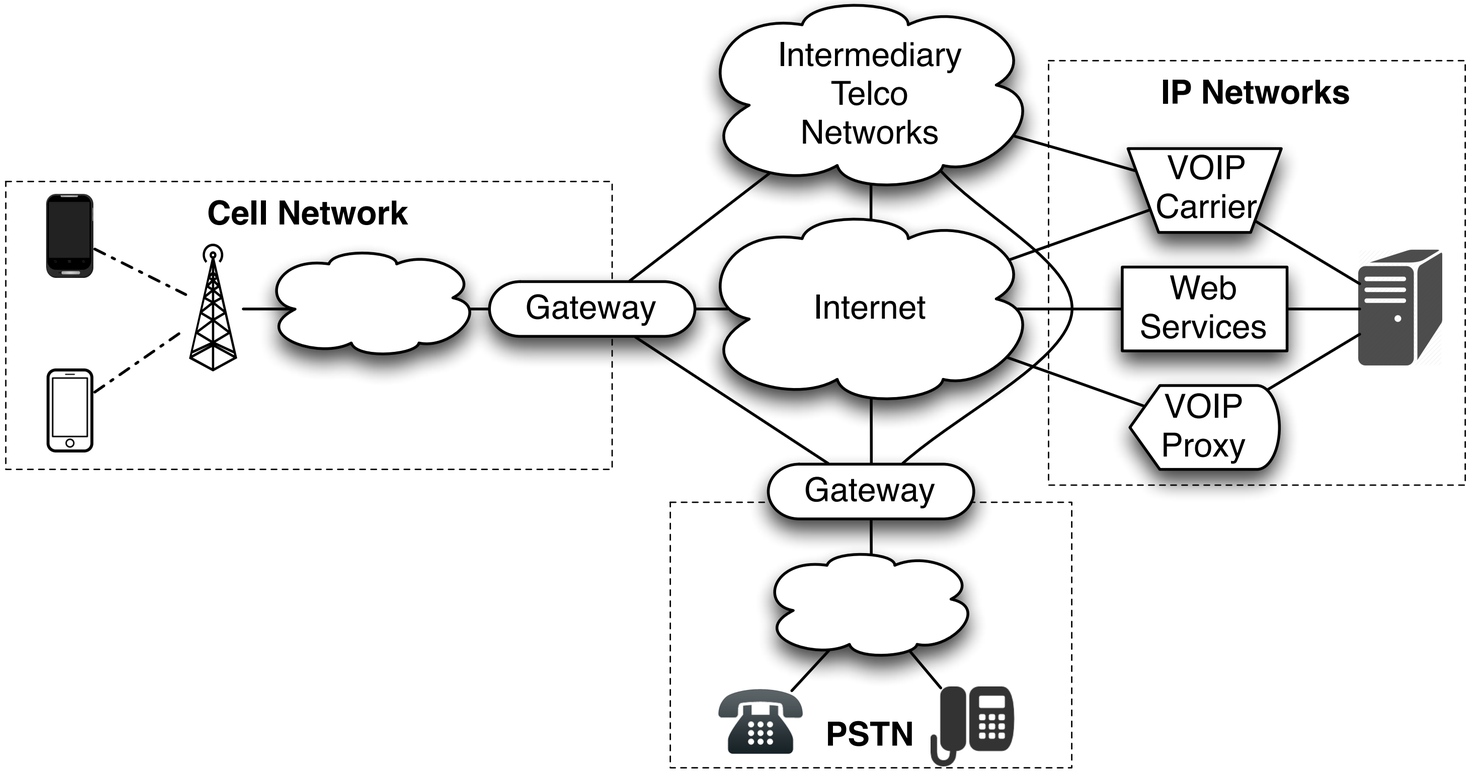
But they shouldn't be

- In general, neither carriers nor end users can provide guarantees about who is on a call.
- Faking a Caller ID is easy



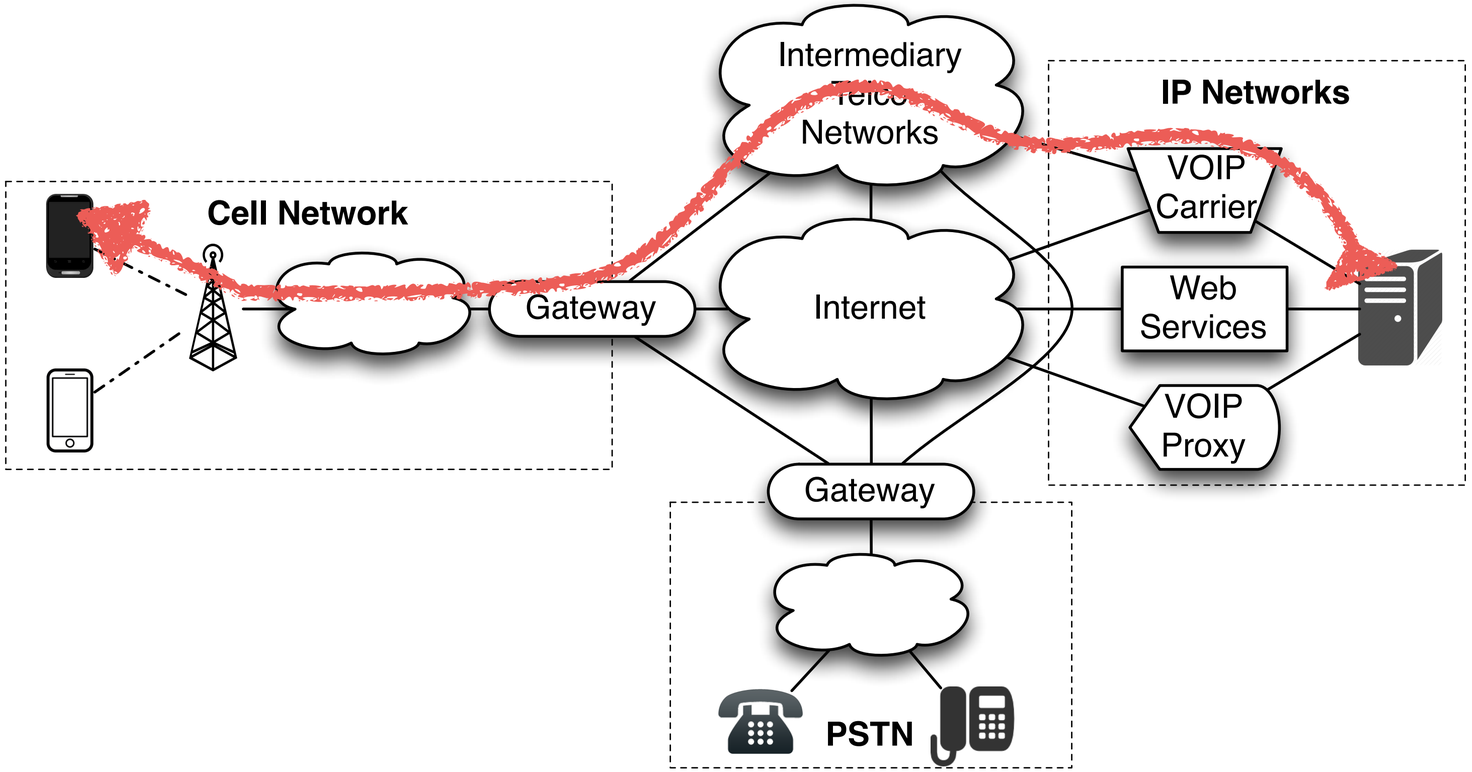
- What is at the root of this problem?

Modern Telephony Systems



Modern Telephony Systems



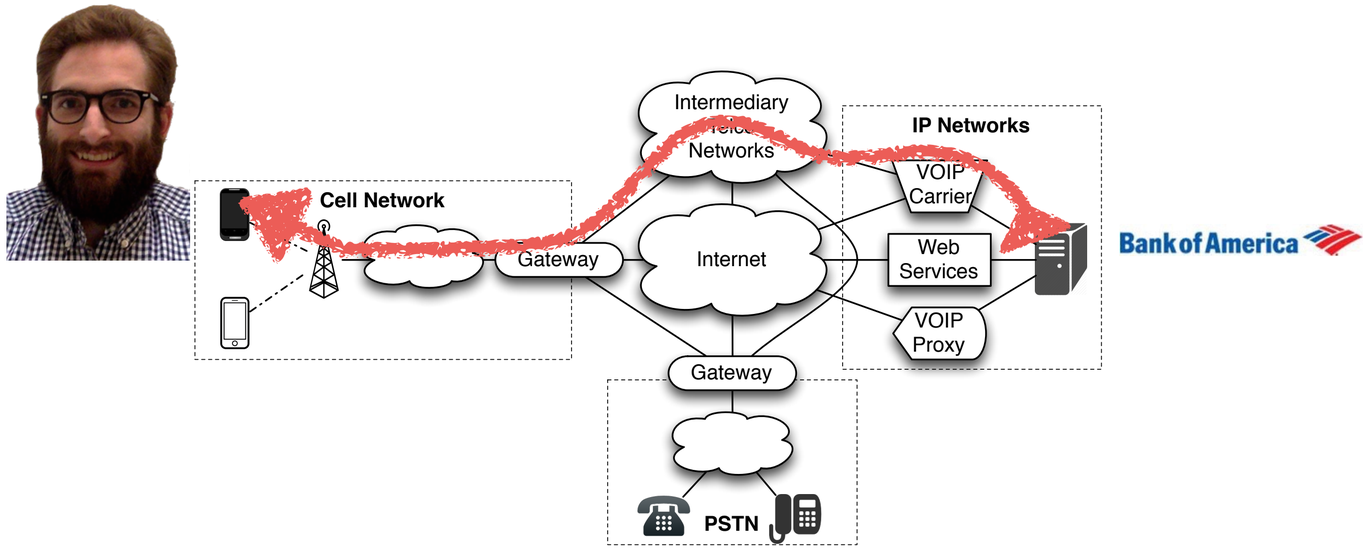


Bank of America

What happens to the call

Along the way:

1. Identity is asserted, not attested
   - No authentication of ID
2. Signaling protocols change
   - ID assertion is not easily fixed
3. Audio compression changes



All networks transmit voice — but they don't always share a way of transmitting data

Web Lessons for Telephony

- We saw similar authentication problems in the early days of the web
- SSL/TLS was developed largely in response to this problem
  - Cryptographic verification of *well-known parties* became widely possible*

- **This paper brings end-to-end explicit authentication to all phone calls**
  - Authloop : Phones :: TLS : Web
  - Can I get some guarantee that the real Bank of America is calling me?

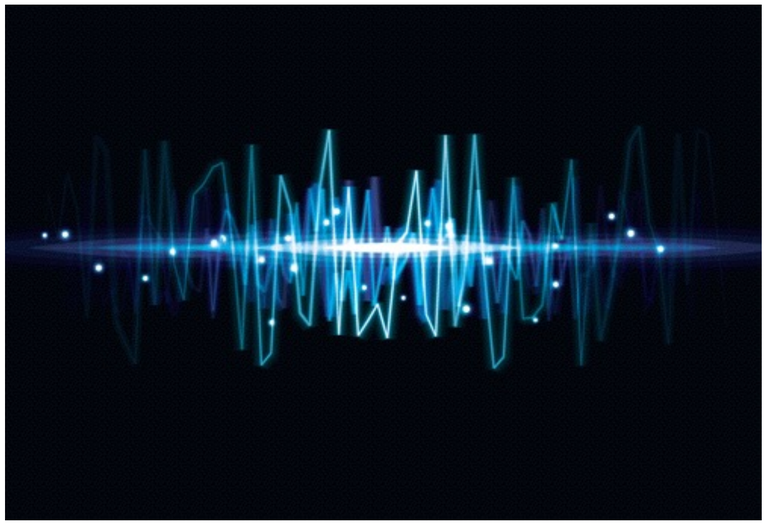


Authloop

- **Authloop authenticates calls cryptographically & end-to-end for the *existing phone network* through the voice channel**
  - *Authenticates calls*
  - *Cryptographically*
  - *End to end*
  - *Existing phone network*
  - *Voice channel*
- *Note: Many apps provide authenticated VoIP, but they only authenticate VoIP calls*

Two problems

- Authloop needs two things:
  - A way to send data through the voice channel
  - An secure, efficient authentication protocol



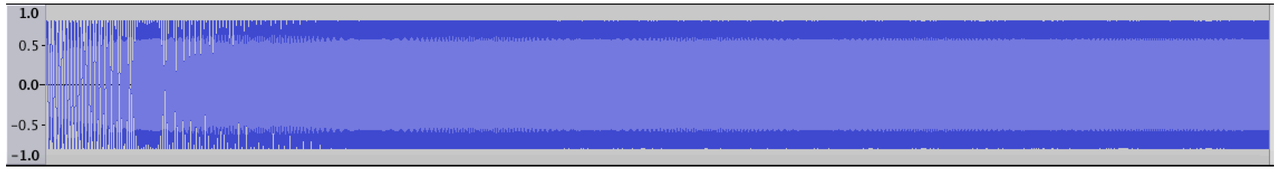


What about a modem?

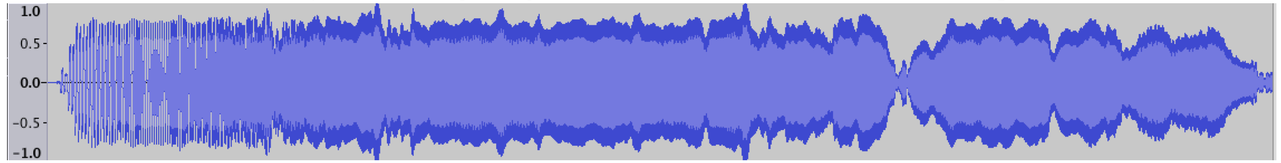


The Problem: Modern Codecs

- Modern codecs make high-fidelity, low bitrate audio possible.
- But these codecs make the transmission of anything other than human voices *completely unreliable*



a) 1-second chirp sweep from 300 - 3300 Hz before AMR-NB encoding

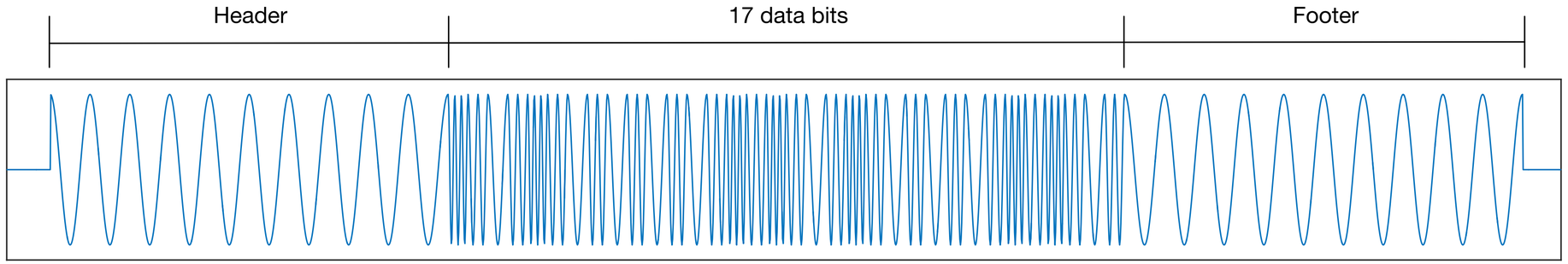


b) 1-second chirp sweep from 300 - 3300 Hz after AMR-NB encoding

- Almost all of the traditional digital comm. techniques go out the window
  - Amplitude not preserved (~~ASK~~, ~~QAM~~, ~~TCM~~, & ~~PCM~~)
  - Phase discontinuities not preserved (~~PSK~~ & ~~QPSK~~)

Codec Agnostic Modem

- Solution: Encode data as **changes** in frequency
  - 3 frequencies available: 1,2,3 kHz
  - Use Manchester encoding (1=> 10, 0=>01) to limit runs of values
  - Transmit 20ms constant-frequency header to open VAD gates



Punchline: approximately 500 bps goodput in the best case

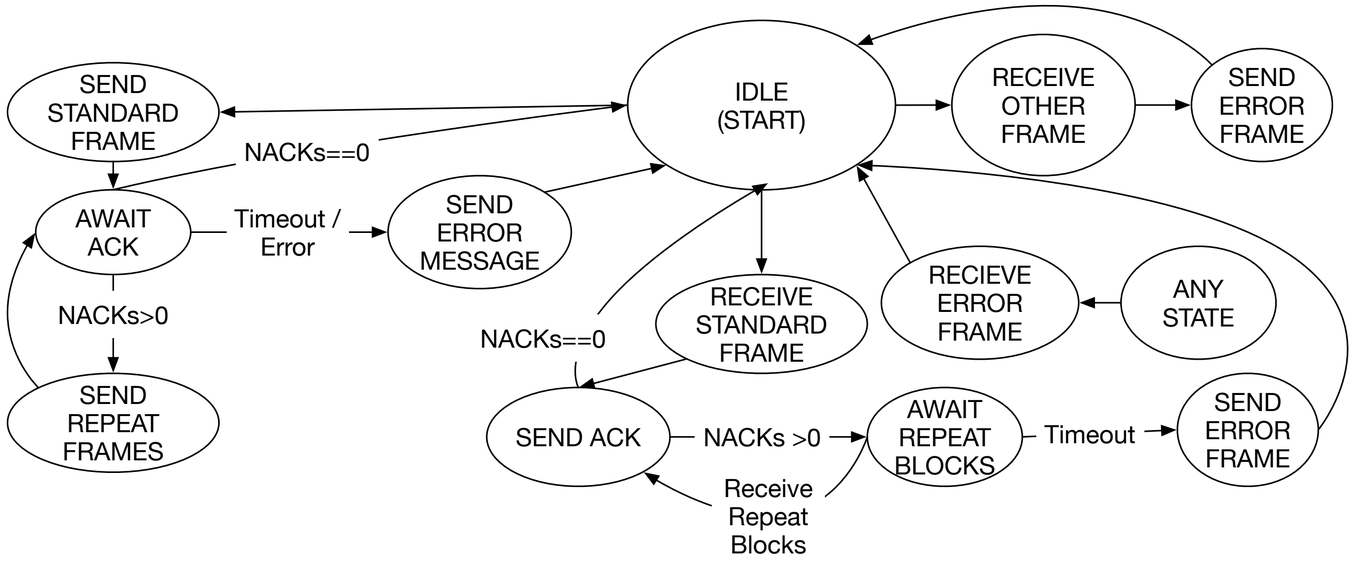
Modem Evaluation

- We measured bit error rate 100 frames containing 2000 bits each
- The modem provides a low bit error rate across codecs for:
  - Landline (G.711)
  - Cellular (AMR-NB)
  - VoIP (Speex)

Codec	Avg Bit Error	Std.Dev
G.711	0.0%	0.0%
AMR-NB	0.3%	0.2%
Speex	0.5%	5%

Link Layer

- We need a link layer with as little overhead as possible
- While still detecting common bit errors on retransmission



Link Layer Performance

- There's a 1.3% chance that the L2 won't detect a bit error
- Actual goodput heavily dependent on BER
- Fortunately, performance is good for the error rates we see in practice

Bit Error Rate	Transmission Time	Goodput
0.1%	4.086 s	490 bps
1%	6.130 s	326 bps
2%	11.652 s	172 bps

Strawman: SSL/TLS

- With a reliable data channel in place, why can't we use TLS?
- **Problem**: Using a standard TLS handshake is too slow

Site Name	Total Bits	Transmission Time at 500 bps
Facebook	41,544	83.088 s
Google	42,856	85.712 s
Bank of America	53,144	106.288 s
Yahoo	57,920	115.840 s
Average	48,688	97.232 s

- **Solution**: A protocol with the guarantees of TLS 1.2 …
  But a fraction of the bandwidth requirement

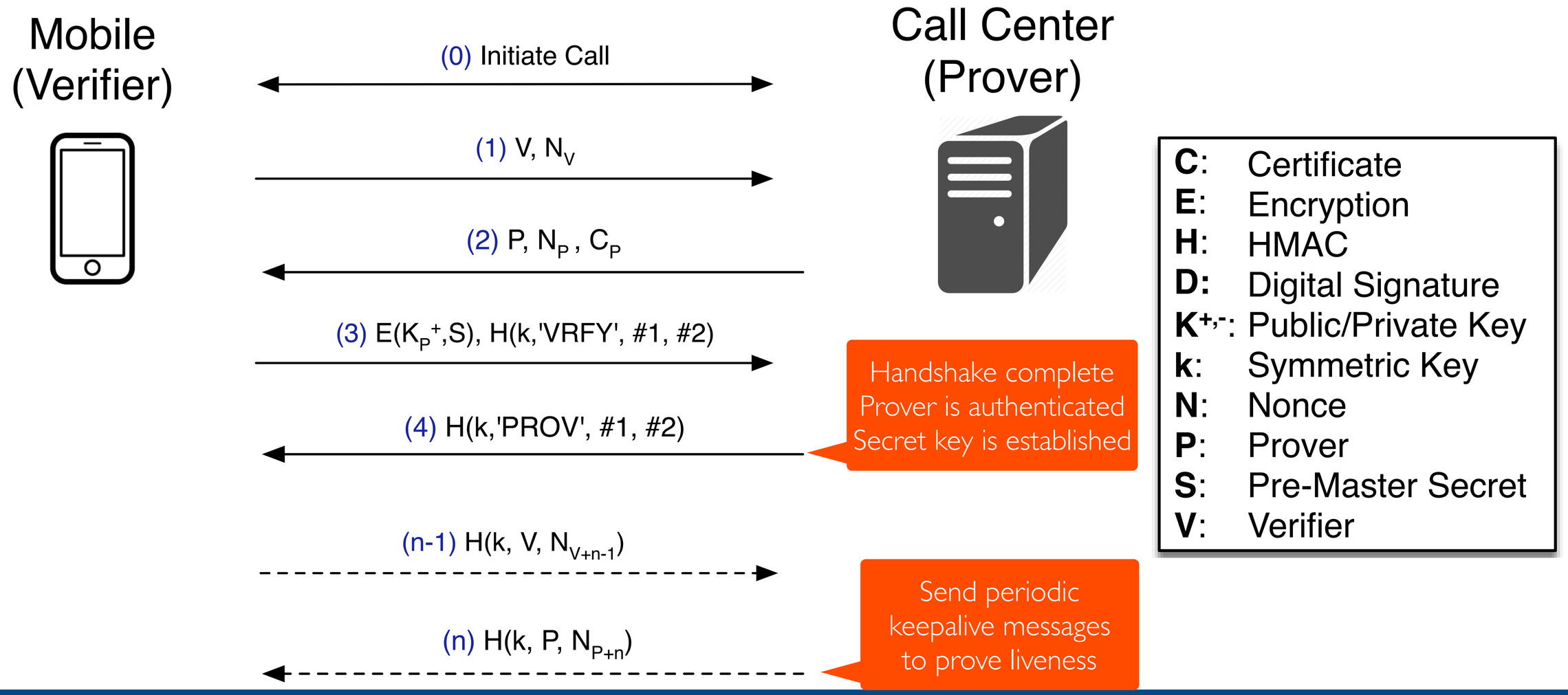
Slimming Down TLS

- **Things that we need:**
  - Cryptographic attestation of identity (i.e. a PKI)
  - Freshness and Liveness tests
  - Shared secret establishment
- **Things we can live without:**
  - RSA and very long HMACs (80bits)
  - Cipher Agreement
  - TLS Record Protocol



The Protocol

Protocol is based on TLS 1.2 Key Transport, verified with ProVerif



Great… But Does It Work?

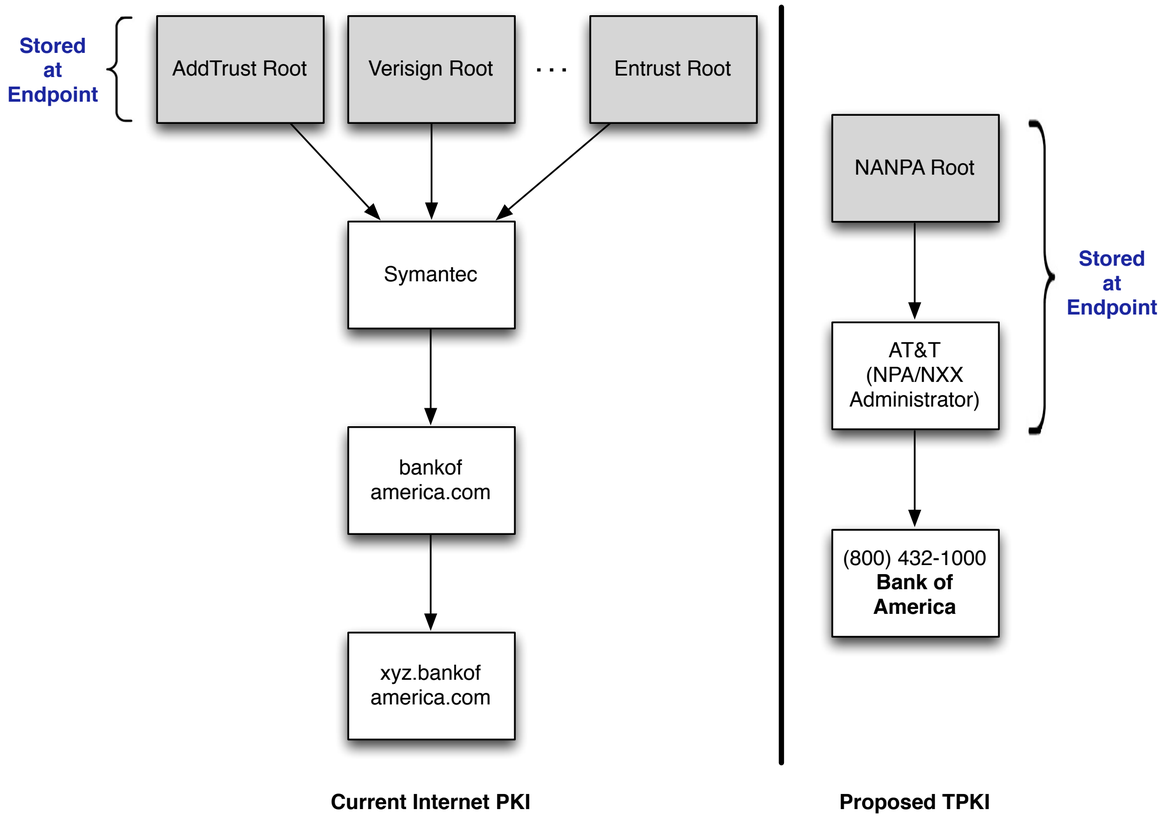
- On average it takes ~9 seconds to do a full handshake.
- If we can cache certificates, we can do it in ~5 seconds

Codec	Cached Certificate	Certificate Exchanged
G.711	4.463 s	8.279 s
AMR-NB	5.608 s	10.374 s
Speex	4.427 s	8.279 s
Average	4.844 s	8.977 s

- Network transmission accounts for 99% of this time — only 50 ms of computation

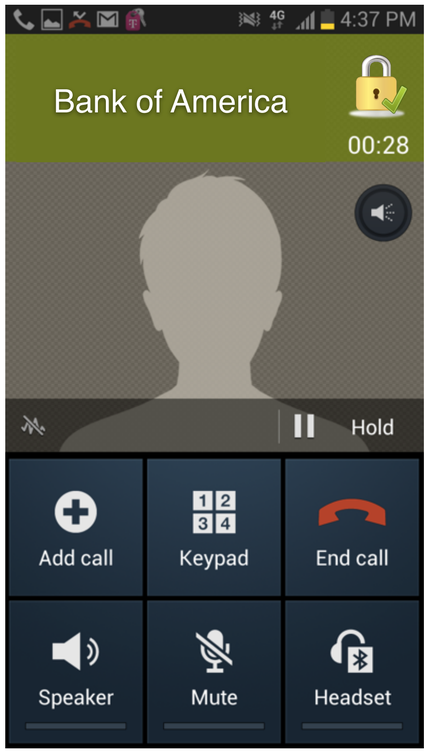
Telephony PKI

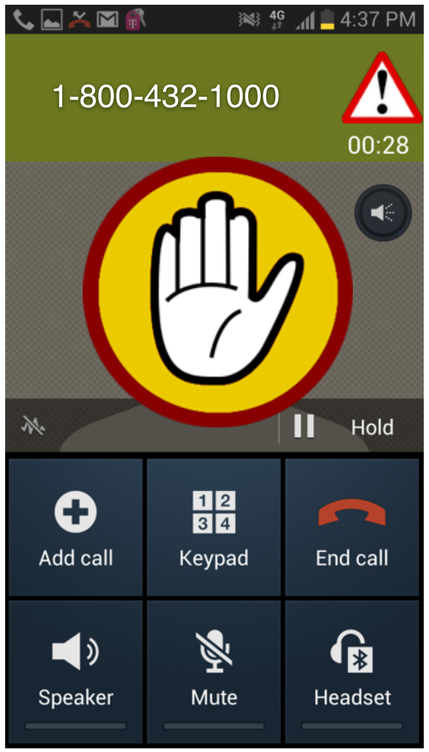
- One of the major problems in the Internet is confusion over valid CAs and identity bindings.
- Telephony naturally lends itself to a singly rooted system using the North American Numbering Plan (NANPA).
  - Carriers are publicly allocated blocks of numbers, so assignments are based on authority.
- Certs for every carrier can be stored in ~100 KiB.
  - No more long, ambiguous certificate chains!



Summing up…

- Telephones are used for our most trusted communications, but one can't be sure who is calling
- **Authloop** solves this problem by providing end-to-end *cryptographic* authentication







BradReaves.net