Distance bounding
Why distance bounding?

Authentication alone may not be sufficient
• physical access to buildings etc.
  - watch out for relay attack

Two main types of attack
• “Mafia Fraud”
• “Terrorist Fraud”
Authentication without distance checking
• Correct response
• from legitimate tag
• ... but attacker gets access!

Famous urban myth: Mig-in-the-middle attack
Relay attacks: Terrorist Fraud

More powerful than Mafia fraud:
- **legit device does not have to be tricked,**
- **... does not have to follow the protocol**
- **... can provide more info than just response**
Countermeasures

What to do against relay attacks?

• Ask the prover where he is
  - but he could be lying

• Signal strength
  • can be spoofed

• Measure the distance to the prover
  - “distance bounding”
  - nothing travels faster than light \( c = 2.99792458 \cdot 10^8 \text{ m/s} \)
  - infer distance from traveling time of signal

300 meters per microsecond
Distance bounding

Demand response within time $t_{\text{max}}$

- travel time to distance $x_{\text{max}}$ and back
- allow some “slack” time for computations
- dist. measurement & proof of knowledge at the same time

\[
t_{\text{max}} = 2 \frac{x_{\text{max}}}{c} + t_{\text{slack}}
\]

\[
x_{\text{spoofable}} = \frac{1}{2} c t_{\text{max}} = x_{\text{max}} + \frac{1}{2} c t_{\text{slack}}
\]

has to be very small
**Distance bounding: practical problems**

\( t_{slack} \) must be very small

• no (heavy) computations
  - addition lasts too long
  - *but still cryptographic challenge-response protocol!*

• delays inside prover device become problematic
  - missed cycles, bus speed, etc

• no error correction
  - we have to live with transmission errors
Solving the practical problems

• no (heavy) computations
  - split protocol into slow and quick phase
  - prover creates LUT in slow crypto phase
  - verifier: unpredictable selection from LUT in quick phase
• delays inside prover
  - LUT sitting right “next to” emitter
• no error correction
  - decide afterwards if there were transmission errors
The Brands-Chaum protocol

Prover
\[ m \in_R \{0,1\}^n \]

Verifier
\[ c \in_R \{0,1\}^n \]

Commit(m) →

[begin rapid phase]

\[ r_i = c_i \oplus m_i \]

Start timer

\[ c_i \]

Stop timer

\[ r_i \]

[end rapid phase]

\[ s = \text{sign}(c || r) \]

s, Open Commit →

Verify timing

Verify Commit

Verify \( r = c \oplus m \)

Verify signature
Do you see a problem with the Brands-Chaum protocol?
**Swiss Knife protocol (2008)**

**Reader has DB \( \{ID, x\} \)**

- Random \( N_A \);
- random \( d \) (Hamm.weight \( m \))

**Tag \( \{ID, x\} \)**

- Random \( N_B \)

\[ Z^0 = f_x(C_B, N_B); \quad Z^1 = Z^0 \oplus x; \]

For \( i = 1 \) to \( m \) \{ \( j = \) index of next 1 in \( d \); \)

\[ R_{0i}^0 = Z_j^0; \quad R_{1i}^1 = Z_j^1 \}

**Rapid bit exchange**

- Random bit \( c_i \); start clock
- Stop clock; store \( \Delta t_i \)

**For \( i = 1 \) to \( m \)**

- \( c_i' \)

\[ r_i = \begin{cases} R_{0i}^0 & \text{if } c_i' = 0 \\ R_{1i}^1 & \text{if } c_i' = 1 \end{cases} \]

**Find matching \( \{ID, x\} \) in DB;**

- compute \( R^0, R^1 \);

- \( \text{err}_c = \#\{ i: c_i' \neq c_i \} \);

- \( \text{err}_r = \#\{ i: c_i' = c_i \land r_i \neq R^{c_i} \} \);

- \( \text{err}_t = \#\{ i: c_i' = c_i \land \Delta t_i > \Delta t_{\text{max}} \} \);

- if \( \text{err}_c + \text{err}_r + \text{err}_t \geq T \) reject;

- \( t_A = f_x(N_B) \)

**Check \( t_A \)**

- \( t_B = f_x(c_1', \ldots, c_m', ID, N_A, N_B) \)
Why is the Swiss Knife protocol secure against the Mafia Fraud?

Is it secure against the Terrorist Fraud?
Still too slow!

State of the art hardware:
• analog $\rightarrow$ digital conversion: 50 ns
• all conversion steps together: 170 ns (26 meters)

Only analog processing is fast enough!
Analog challenge-response

Rasmussen & Čapkun 2010
• Adaptation of Brands-Chaum
• Single register R.
• CRCS: Challenge Reflection with Channel Selection.

Challenge: unpredictable signal $c(t)$ at frequency $f_c$

Response: reflection of $c(t)$ at shifted frequency
Challenge Reflection with Channel Selection

< 1 nanosecond !
Analog version of Swiss Knife rapid phase

**Challenge:** $c_i(t)$ at frequency $\omega_c$ ($\omega_0$ or $\omega_1$).

**Response:** reflection of $c_i(t)$ at shifted frequency; shift depends on $R_{ci}$.

$$\omega_{\text{response}} = \omega_c + (2 R_{ci} - 1) \omega_{\Delta}$$