Distance bounding overview
Distance bounding overview

- **Mafia fraud**: an adversary tricks a verifier into thinking that a prover is near, by establishing a relay link between them.

- **Distance fraud**: the prover itself is malicious, and tricks the verifier into thinking to be near.
Distance bounding overview

• A *distance bounding protocol* permits us to establish a *secure upper bound* \((D)\) to the distance between a “prover” and a “verifier”:

\[
d \leq D
\]

• The basic idea is to precisely measure the *round-trip time* between two unpredictable messages (a challenge and a response)
Distance bounding overview

The prover is surely in this circle

\[ D \]

\[ D_{\text{max}} \]
Brands-Chaum protocol (type I)

It resists only against mafia fraud
Brands-Chaum protocol (type II)

Prover

Verifier

$N$ random bits: $c_i$

random bit-string: $salt$

secret key: $k$

$m = a_1|b_1|...|a_N|b_N$

$b_i = a_i \oplus c_i$

$hash(c_1|...|c_N|salt)$

$x \cdot N$ times:

$public key: k^{-1}$

$N$ random bits: $a_i$

$salt, sign_k(m)$

It resists against both mafia fraud and distance fraud
Brands-Chaum protocols

• Type I:
  • Adversarial success probability (mafia fraud):
    \[ P_{\text{adv}} = (1/2)^N \]

• Type II:
  • Adversarial success probability (mafia and distance frauds):
    \[ P_{\text{adv}} = (1/2)^N \]
Hancke-Kuhn protocol

It resists against both mafia fraud and distance fraud.
Hancke-Kuhn protocol

- Adversarial success probability (mafia fraud):
  - Double-chance guessing attack
  - Overclocking attack

\[ P_{\text{adv}} = \sum_{i=N_{\text{accept}}}^{N} \binom{N}{i} \left( \frac{3}{4} \right)^i \left( \frac{1}{4} \right)^{N-i} \]

- Adversarial success probability (distance fraud):

\[ P_{\text{adv}} = \sum_{i=N_{\text{accept}}}^{N} \binom{N}{i} \left( \frac{3}{4} \right)^i \left( \frac{1}{4} \right)^{N-i} \]

- With \( N=128 \) and \( N_{\text{accept}}=124 \): \( P_{\text{adv}} = 10^{-12} \)
Frame-based distance bounding

- Medium range communication (20-30 meters): we cannot send single bits
- We use the same protocols (Brands-Chaum, Hancke-Kuhn)
- Instead of performing N single-bit rounds, we perform a single round with an N-bit frame
Frame-based distance bounding

It resists against mafia fraud only
Frame-based distance bounding

Prover

Verifier

It resists against both mafia fraud and distance fraud
Distance bounding implementation
Secure positioning
Problem type

- Secure *positioning* (properly said): to securely measure the position of a device
- Secure *position verification*: to verify that a (previously measured) position is actually true
Positioning method types

- **Range-dependent**: based on the *ranging operation* (the measurement of a distance)
  - Very precise
  - Expensive (dedicated hardware for ranging)

- **Range-independent**: based on higher-level information (signal strength, beacon reception, etc.)
  - Poorly precise
  - Cheap (no dedicated hardware)
Multilateration

• Range-based positioning method
• Based on the measurement of 3 (or more) distances from the *target node* to 3 (or more) anchor nodes
Multilateration
Multilateration

- In presence of ranging errors: *least-squared-error solution* (LSE)
Multilateration

- $d_i$ is the distance from anchor node $V_i$
- $d'_i$ is the measured distance from $V_i$
- $X_P$ is the position of the target node
- $X'_P$ is the measured position of the target node
Multilateration

• Without ranging error (exact solution):

\[\begin{align*}
|X_{V_1} - X_P'| &= d_1' \\
|X_{V_2} - X_P'| &= d_2' \\
|X_{V_3} - X_P'| &= d_3'
\end{align*}\]

• With ranging error (least-squared-error solution)

\[\min \sum \delta_i^2\]

\[\begin{align*}
|X_{V_1} - X_P'| - d_1' &= \delta_1 \\
|X_{V_2} - X_P'| - d_2' &= \delta_2 \\
|X_{V_3} - X_P'| - d_3' &= \delta_3
\end{align*}\]
Multilateration
Multilateration

- The residuals give an indirect estimation of the positioning imprecision
- If the residuals are high, the positioning is imprecise (the contrary could not be true)
Multilateration spoofing

Distance enlargement

\[ d'_1 > d_1 \]

Distance reduction

\[ d'_3 < d_3 \]

\[ d'_2 = d_2 \]
Verifiable multilateration

- *Idea*: perform ranging operations via wireless distance bounding protocols
- Distance reduction is *impossible*
- Distance enlargement is still possible
  - *Jam-replay* (jamming a response and replaying it)
  - *Overshadow* (replaying a response with much more power)
Verifiable multilateration

\[ d' > d \]

\[ d' = d \]

\[ d' < d \]

\[ X_P \]

\[ X'_P \]
Verifiable multilateration

\[ d'_1 > d_1 \]
\[ d'_2 > d_2 \]
\[ d'_3 > d_3 \]
Verifiable multilateration

- Accept a position only if it is inside the polygon formed by the anchor nodes \((\text{in-polygon test})\)

- Spoofing a position inside the polygon \textit{always} requires a distance reduction
Verifiable multilateration

- Case of "inside-inside" spoofing

- Distance reduction against $V_3$ (impossible)
Verifiable multilateration

- Case of “outside-inside” spoofing

- Distance reduction against $V_2$ (impossible)
Verifiable multilateration

- The adversary can spoof the position only by means of distance enlargement
Verifiable multilateration

- Accept a position only if it produced low residuals ($\delta$-test)
Verifiable multilateration

Complete algorithm:

1. Determine the list of anchor nodes inside the power range of the target
2. For each anchor node, perform distance bounding
3. Compute the position by means of least-squared-error problem
4. If one residual is greater than a threshold $\delta_{max}$, then reject the position ($\delta$-test)
5. If the position is not inside the polygon of the anchor nodes, reject the position (in-polygon test)
6. Otherwise, accept the position
The coverage area is smaller than (classic) multilateration
Coverage

- Best way to deploy anchor nodes (*hive deployment*)
Coverage

- Best way to deploy anchor nodes (*hive deployment*)
  
  \[
  \begin{array}{cccccccccccc}
  \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
  \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
  \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
  \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
  \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
  \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
  \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
  \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
  \end{array}
  \]

  Repeat the scheme
Security analysis

- Verifiable multilateration has the same security level of the employed distance bounding
- Case of *external adversary*: use a distance bounding resistant against mafia fraud (e.g. Brands-Chaum type I)
- Case of (single) *dishonest target node*: use a distance bounding resistant against mafia and distance frauds (e.g. Hancke-Kuhn)
- Case of *multiple* dishonest target nodes?
Colluding-internals attack

- $P_2$ attacks $V_2$ and $V_3$
- $P_1$ attacks $V_1$
- Verifiable multilateration does not resist against colluding dishonest targets
Simultaneous verifiable multilateration

- Instead of $N$ distance boundings: a single intertwined distance bounding
- Intertwined distance bounding: multi-party distance bounding ($1$ prover, $N$ verifiers)
  - A challenge for each verifier
  - The challenges arrive simultaneously to the prover ($N$ wireless channels)
  - A single (broadcast) response from the prover
  - The response depends on all the challenges
Intertwined distance bounding

$$b = f_{cr}(a_{V1}, a_{V2}, a_{V3}, c)$$

Brands-Chaum style:
$$b = a_{V1} \oplus a_{V2} \oplus a_{V3} \oplus c$$

Hancke-Kuhn style:
$$b = \begin{cases} m & \text{if } (a_{V1} \oplus a_{V2} \oplus a_{V3}) = 0 \\ n & \text{if } (a_{V1} \oplus a_{V2} \oplus a_{V3}) = 1 \end{cases}$$
Security analysis

- The verifiers send the challenges in such a way they arrive contemporaneously at the *supposed position* $P'$.  
- $P_2$ cannot perform the enlargements, because he didn't receive the $V_1$'s challenge yet.
Security analysis

- The colluding internals attack is *still possible*, but in fewer situations.
- It generally needs more colluders.
Security analysis

- Simultaneous verifiable multilateration only *mitigates* the colluding-internals attack

- *Theorem (Chandran-Goyal-Moriarty-Ostrovsky):* if the number of colluders is equal to (or greater than) the number of verifiers, no time-of-flight positioning is secure
Requirements for the intertwined distance bounding

- The system must already know a *supposed position* $P'$ (secure position verification)
  - The target itself declares it
  - Or it can be measured with an insecure method (like classic multilateration or GPS)
- The anchor nodes must be perfectly synchronized (with nanosecond precision)
  - Synchronization via *cable*: quite expensive
  - Synchronization via *wireless*: possibly insecure (an adversary can attack the synchronization protocol)
Trusted-hardware distance bounding

- An alternative way to avoid dishonest provers is to use *trusted hardware* for implementing distance bounding.
- The correct execution of the protocol is assured by the trusted hardware.
- A prover (or a set of colluding provers) *cannot act dishonestly*.
- We can use simpler distance bounding protocols, like Brands-Chaum type I (no distance fraud is possible)
Trusted-hardware distance bounding

- The protocol is implemented in hardware

- The key (endorsement key) is created at manufacture time and stored in hardware

- Nobody knows the key except for the trusted hardware module
SeRLoc

- Secure Range-independent Localization
- Nodes are *not* equipped with ranging hardware (cheaper)
- Target nodes are *trusted*, they determine their own position
- The anchor nodes periodically send authenticated beacon packets
- Target nodes determine their own position by listening to the beacon packets
SeRLoc

• The beacon packets are protected against jamming and
Secure GPS
GNSS

• GNSS = Global Navigation Satellite System
• Examples:
  • GPS (USA, global)
  • GLONASS (Russia, global)
  • Galileo (UE, under construction)
  • Compass (China, regional, to be expanded to global)
GNSS

- Satellite constellation
- *Pseudo-ranging* operation from satellite to earth
- The satellite periodically broadcasts a navigation message
- The GPS receiver measures the instant of arrival
The satellites are synchronized each other (atomic clocks)

The ground GPS receiver and the satellites are not synchronized (sky-ground clock difference: $\Delta t_{S-G}$)

The GPS receiver knows the satellite position ($X_S$) and time ($t_S^{(tx)}$) when the satellite broadcasted the message

$|X_S - X_G| = (t_G^{(rx)} - t_S^{(tx)} - \Delta t_{S-G}) \cdot c$

3 unknowns (x, y, z)

1 unknown

Pseudo-ranging result
GNSS

- Four pseudo-rangings with four different satellites

\[
\begin{align*}
|X_{S_1} - X_G| &= (t_{G_1}^{rx} - t_{S_1}^{tx} - \Delta t_{S-G}) \cdot c \\
|X_{S_2} - X_G| &= (t_{G_2}^{rx} - t_{S_2}^{tx} - \Delta t_{S-G}) \cdot c \\
|X_{S_3} - X_G| &= (t_{G_3}^{rx} - t_{S_3}^{tx} - \Delta t_{S-G}) \cdot c \\
|X_{S_4} - X_G| &= (t_{G_4}^{rx} - t_{S_4}^{tx} - \Delta t_{S-G}) \cdot c
\end{align*}
\]

- The pseudo-rangings are affected by an error
  - They do not intersect in a single point
  - Least-square-error solution is computed
Civil and military GNSS

- Most of GNSS system (e.g. GPS) uses two types of navigation signals:
  - Civil navigation signal
  - Military navigation signal
- The military navigation signal uses spread-spectrum modulation with a secret spreading code
  - It is hard to receive, to synthesize, or to jam military signals unless the spreading code is known
GPS jamming/spoofing

- **GPS jamming**: to disturb the bandwidth on which the (civil) navigation signals are transmitted, in such a way to interrupt the navigation service

- **GPS spoofing**: to synthesize false (civil) navigation signals, in such a way to deceive the navigation service
Truck stealing

- Suppose a truck is carrying valuable goods (gold, etc.)
- The truck is protected by a satellite anti-theft system
  - GPS receivers + cellular connection to an operations center (usually by SMSs)
- The driver has also a “panic button” with which he can send an alarm
Truck stealing

Operations center (Police, etc.)

Time: $T$

Position: $X$

Panic state: $P$

Secret key: $k$

$T, X, P, \text{sign}_k(T, X, P)$

$T, X, P, \text{sign}_k(T, X, P)$

...
Truck stealing

- If the signature is bad, an alarm will be raised.
- If no updates are received for more than ten minutes to the police station, an alarm will be raised.
- If the panic-state is “pushed”, an alarm will be raised.
- If an alarm is raised, a police helicopter team will arrive.
Truck stealing

● Buy (or borrow) a GPS signal simulator
  ● For example: Spirent GSS6700 Multi-GNSS Constellation Simulator System
Truck stealing

- Follow the truck and spoof its GPS receiver
- Make the police station believe that the truck has stopped at a service station
- Wait until the truck is far away from its fake position
Truck stealing

• *Make the truck stop!*

• If the driver pushes the panic button, the police helicopters will reach the fake position

• Once you have the control of the truck, disable all the other security mechanisms

Attack performed in Russia, 1999
Boat hijacking

- A boat follows automatically a predefined route
- The route-following is controlled by means of GPS
Boat hijacking

- Follow the boat and spoof its GPS receiver
- Make it believe that it deviated from the route
Boat hijacking

- The control system tries to *correct* the route to the predefined one
- The boat turns left

Fake route

Hijacking!

Attack successfully tested in 2012
(Austin University)
Secure GPS

• Main problems of securing existing (civil) GPS:
  • One-way communication (no distance bounding!)
  • Legacy protocols (GPS messages are not authenticated)
  • Protocol modifications require long deployment times (tens of years)
    – European Galileo will be (probably) authenticated
  • Navigation signals reach earth with very low power
    – It is easy to overshadow them with fake signals
Multi-antenna defense

- **Idea**: equip the GPS receiver with two antennas
- By measuring the *time difference of arrival* (*TDoA*) it is possible to determine the angle of incidence of the signal

\[ \alpha = \sin^{-1} \left( \frac{TDoA \cdot c}{b} \right) \]
Multi-antenna defense

- In the honest case, the received signals have different angles of incidence (one for each satellite)
Multi-antenna defense

- In the adversarial case, the received signals have the same angle of incidence.
- If the angles of incidence are equal, then reject the position measurement.

\[ \alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 \]
Security analysis

- Colluding adversaries could simulate the angles of incidence of several satellites

\[ \alpha_1, \alpha_2, \alpha_3, \alpha_4 \]
Security analysis

- A single adversary equipped with two directional antennas can hit the two receivers with different signals
Security analysis

- In this way, the adversary can spoof the angle of incidence ($\alpha'$) of each simulated satellite.
Security analysis

- The multi-antenna defense is cheap, but protects only against a single point-transmitter adversary
- More sophisticated attacks are successful
  - multiple point-transmitters
  - directional-transmitter
References

  - Only Sections I, II, IV


  - Only Sections I, III, IV