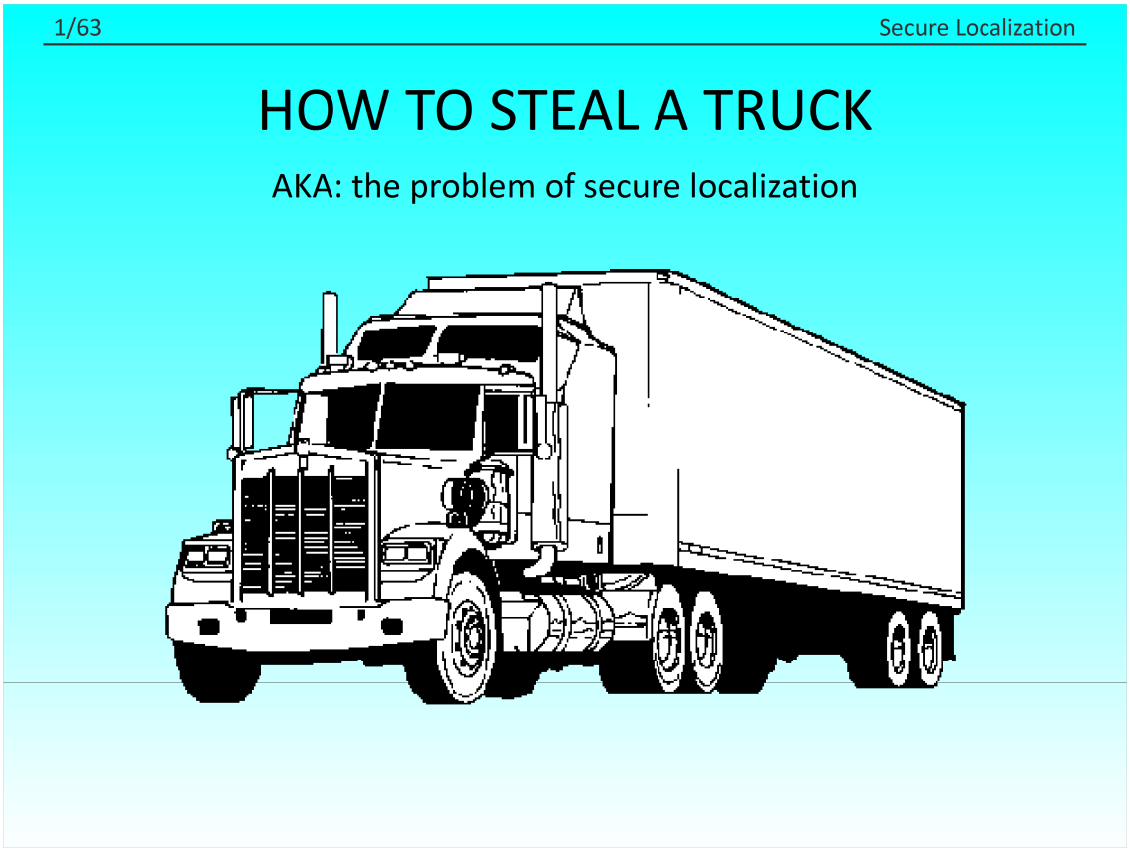
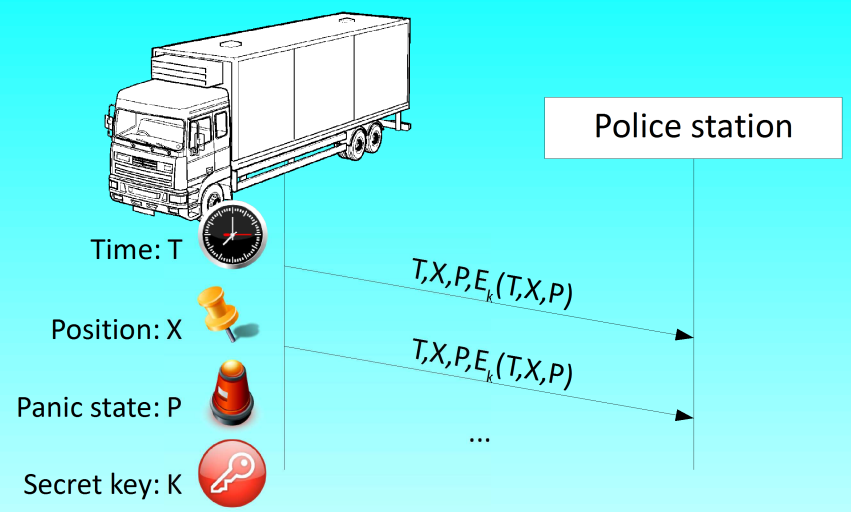


In the first part of the talk, some common location-based security vulnerabilities will be presented. In the second part, the state-of-the-art countermeasures will be described.

GPS tracking for cargoes

- The cargo company BigValues™ protects their trucks with an anti-theft system
- The anti-theft system provides for:
  - a remote GPS tracking
  - an “panic button”

The first vulnerability we tackle is the cargo stealing. Suppose that a company protects its valuable transportation with an anti-theft system. Such a system includes a remote tracking by GPS, and a “panic button” that the driver can push if something go wrong.



- Every ten minutes, the truck sends an authenticated update to the police station

Every 10 minutes, the truck sends (e.g. by LTE), a position update to a remote police station. The position update includes the truck's position, taken from GPS, the current date and time, the state (pushed/not pushed) of the panic button, and a signature.

GPS tracking for cargoes

- 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

If the signature is bad, an alarm is raised. If no updates are received for more than ten minutes, an alarm is raised. If the panic-state is “pushed”, an alarm is raised.

If an alarm is raised for one of these reasons, a helicopter team will arrive at the place within 15 minutes.

How to steal the truck?

Cargo stealing

- Follow these steps:
  1. Gather 1000 dollars and borrow a GPS satellite simulator

Step 1. Borrow a GPS satellite simulator (for 1000$ at month). Such a device simulates GPS signals and is normally used for testing GPS receivers. It can make a GPS receiver measure a fake position.

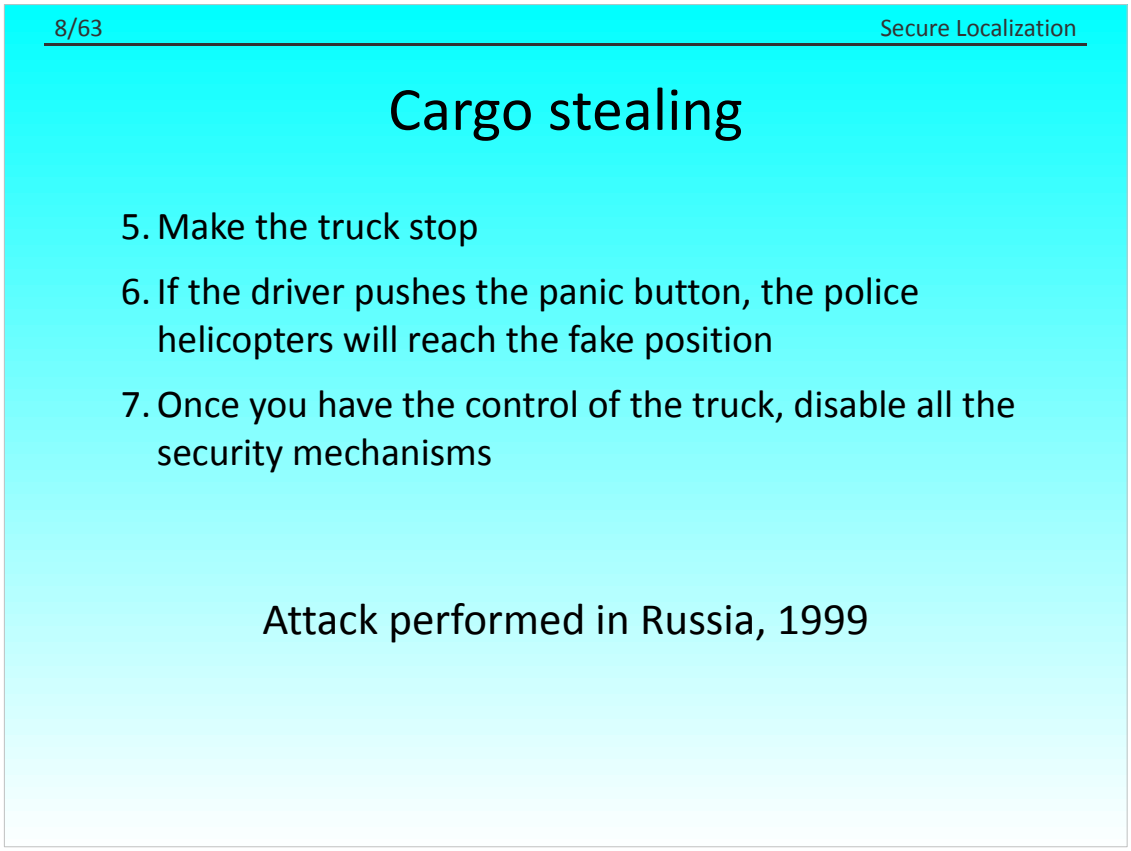
Cargo stealing

2. Follow the truck, overshadowing the legitimate GPS signal with the GPS simulator
3. Make the GPS receiver believe that the truck stopped at a service station
4. Wait until the truck is far away from its fake position

Step 2. Follow the truck and overshadow the legitimate GPS signal with the GPS simulator

Step 3. Make the GPS receiver believe that the truck stopped in a service station, or has taken an alternative way.

Step 4. Wait until the true position and the fake position of the truck are far away enough.



Step 5. Make the truck stop.

Step 6. If the driver pushes the panic button the police helicopters will reach the fake position and find nothing.

Step 7. Once you have the control of the truck, disable all the security mechanisms.

A similar attack has been performed in Russia, in 1999 (Los Alamos NL report).

GPS (in-)security

- Civilian GPS signals are not ciphered neither authenticated
- All the systems which rely on *GPS positioning* are insecure
- All the systems which rely on *GPS synchronization* are insecure

We must never forget that a GPS device is a receiver, rather than a sensor.

(Civilian) GPS signals are not ciphered neither authenticated. As a result, all the systems which rely on GPS positioning service are insecure. The same can be said as well for the systems which rely on GPS synchronization.

How to effectively protect the truck?

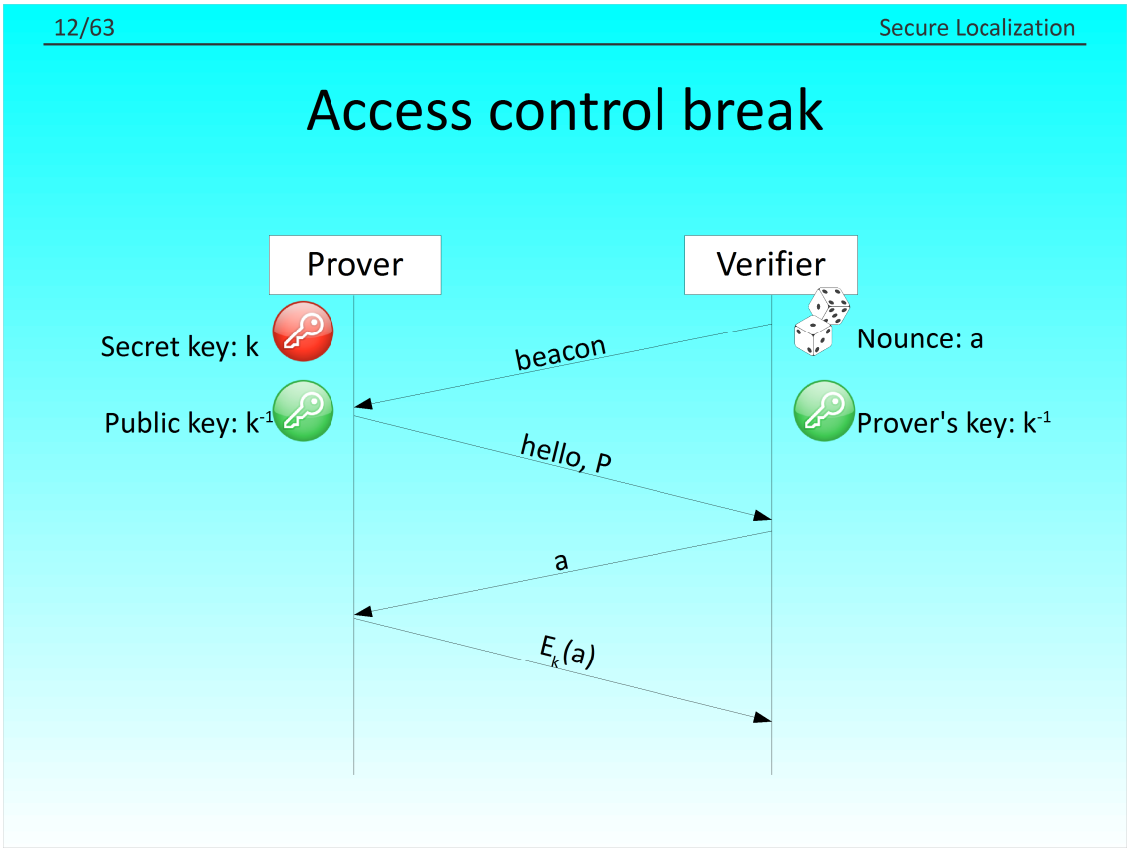
The objective is to securely measure the position of something in presence of an adversary. This problem is not trivial. To solve this, we have first to introduce the relay vulnerability in access control systems and its countermeasures.

Access control break

- The top-secret area contains big military secrets (crashed UFOs, mind-control technologies, etc...)
- The employees access the top-secret area with a personal smart card

- The smart card cannot be cloned (tamper-proofness and asymmetric cryptography)

Suppose we have a top-secret area, which holds military secrets. The employees access the area with a personal smart card. The smart cards (prover) authenticate with the verifier by means of unforgeable asymmetric cryptography. They are tamper-proof and cannot be cloned.



The authentication protocol is shown above. The verifier sends periodically a beacon messages. The prover responds with a hello packet and its ID. Then, the verifier initiates a challenge-response authentication protocol, with a random unpredictable quantity a, which the prover signs with a private key k.

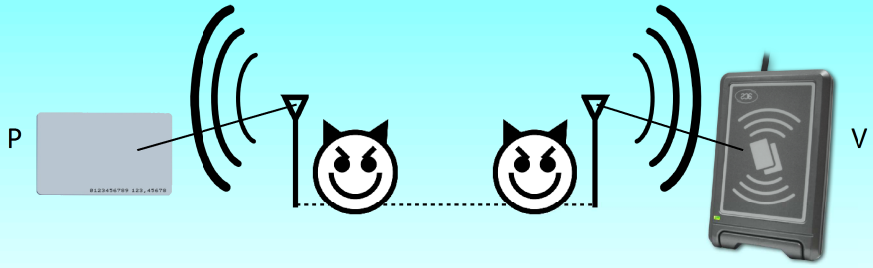
How to enter the top-secret area?

This protocol seems to have nothing to do with location information. Actually, the verifier makes a (possibly false) assumption: that the prover performing the protocol is in the geographic proximity.

The authentication protocol does not give us this security, and a dishonest entity could leverage on this to perform an attack.

Relay attack

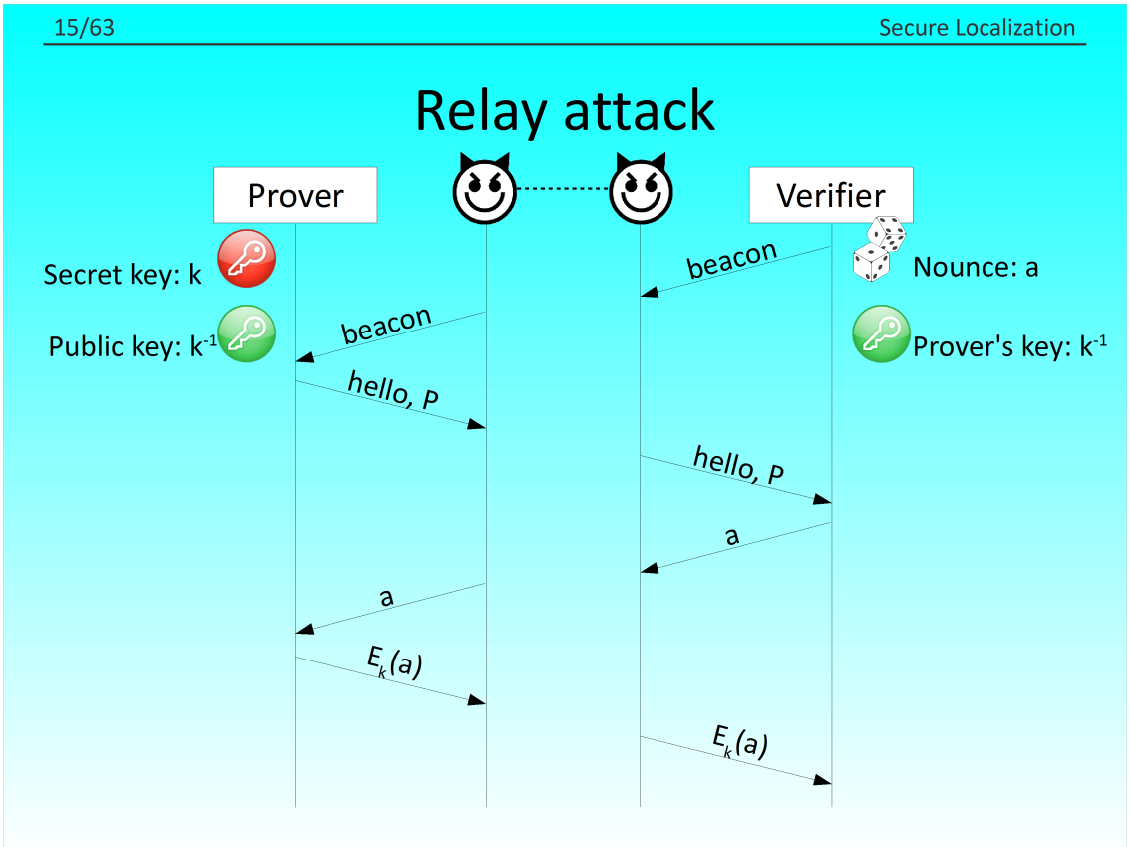
- Build a “relay” link which make a legitimate verifier (V) communicate with a far way legitimate prover (P)



The adversary can enter the secret area by building a “relay” between the verifier and a far away prover. In such a way, two far entities becomes logically “near”, as they can communicate to each other.

The prover

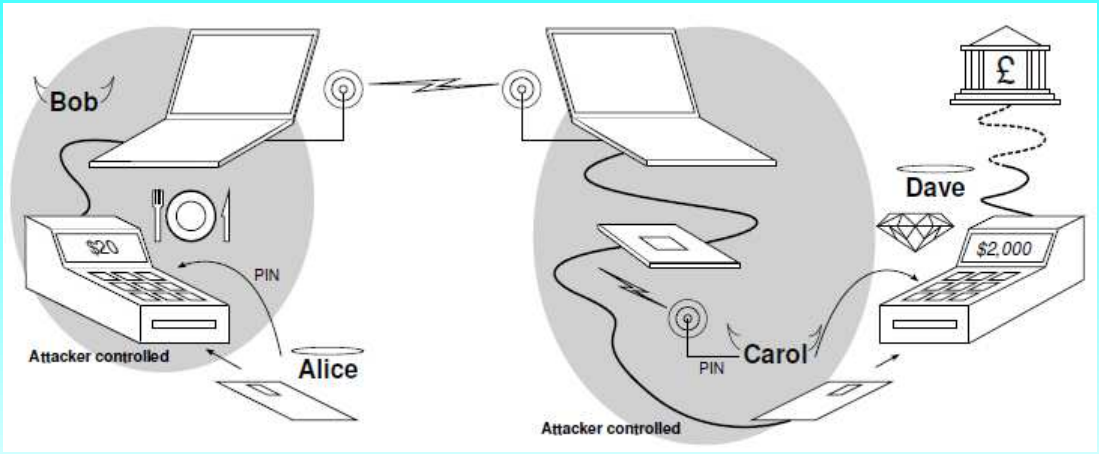
A first adversary's device (V') stands nearby the prover and initiates a communication with it. A second adversary's device (P') stands nearby the verifier and gets the access granted. V' and P' communicates with some kind of link. Either wired or wireless (or simply the Internet).



The adversary replays the messages from P to V and viceversa. In this way, the legitimate prover performs the authentication, but the adversary nearby the verifier enters the top-secret area.

Note that the adversary does not need to understand or modify the content of the messages. Therefore, this vulnerability cannot be solved by solely cryptographic methods.

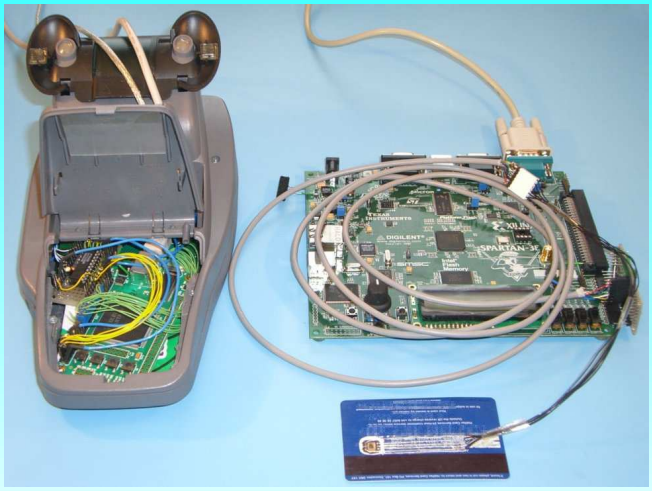
Mafia fraud



The same problem is in electronic payment systems (mafia fraud). Suppose that Carol goes to restaurant and wants to pay with credit card. She does not know that the restaurant ruler is the dishonest Bob. Bob's POS terminal is a modified one. It's not linked to the bank, but rather to Carol, which is Bob's ally. Carol is in Dave's jewelery and is holding a modified smart card, linked to the modified POS terminal of Bob.

When Alice inserts her card to Bob's terminal, Carol inserts her card in Dave's terminal. From the cryptographical point of view, Alice's smart card is authenticating in Dave's POS terminal. Alice thinks that she is paying for a 20$ lunch, but actually she is paying for a 2.000$ jewel that she will never have.

Mafia fraud



2007

This is a 2007 realization of the mafia-fraud relay link. The fake POS terminal seems an ordinary terminal, but the internal circuits have been replaced by a programmed FPGA. The fake smart card seems an ordinary smart card, except for the bind cable, that can be easily hided.

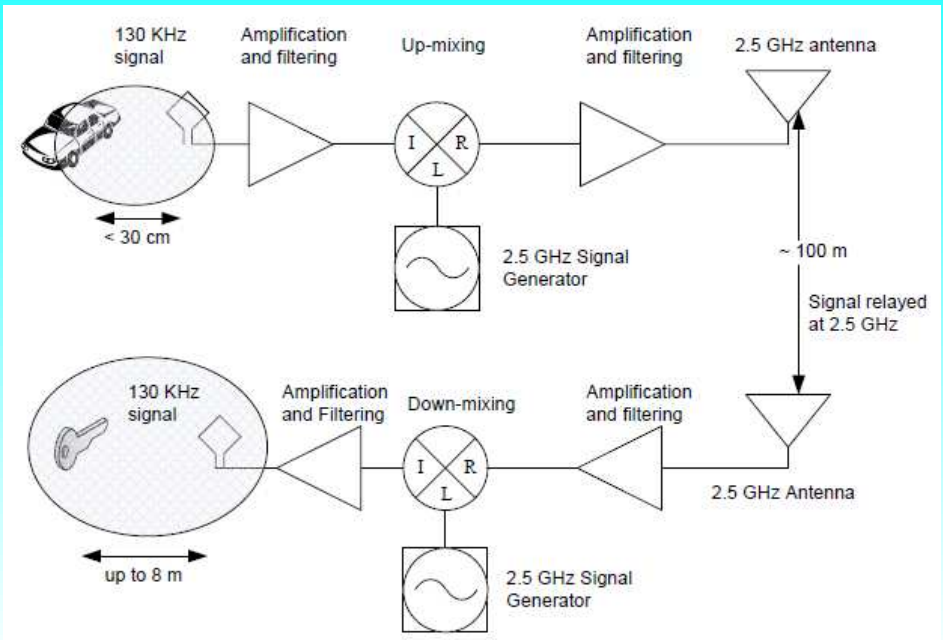
Relay attack on PKES

- Passive Keyless Entry and Start

The same vulnerability is present on PKES systems of the newest car models.

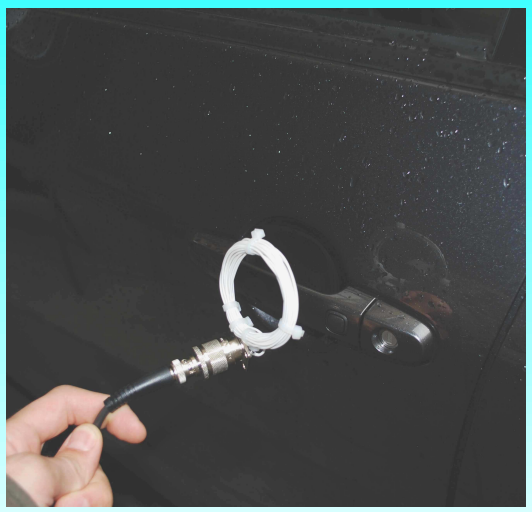
Passive keyless entry and start (PKES) systems, also known as “hands-free keyless entry and start”, permits the owner to open and ignite his car without touching the key. The car simply detects the proximity of the key and performs an authentication protocol.

Relay attack on PKES



These systems are handy for the user, but are handy for the thieves too. A relay hardware can be deployed between the car and the (far away) key.

Relay attack on PKES



2010

This is an implementation example of the relay in 2010.

Relay attack on PKES

Car model	Relay cable					
	7 m		30 m		60 m	
	open	go	open	go	open	go
Model 1	✓	✓	✓	✓	✓	✓
Model 2	✓	✓	A	A	A	A
Model 3	✓	✓	✓	✓	✓	✓
Model 4	✓	✓	-	-	-	-
Model 5	✓	✓	✓	✓	✓	✓
Model 6	✓	✓	A	A	A	A
Model 7	✓	✓	A	A	-	-
Model 8	✓	A	✓	A	-	-
Model 9	✓	✓	✓	✓	✓	✓
Model 10	✓	✓	✓	✓	-	-

✓ Without amplification

A With amplification

\- Not tested

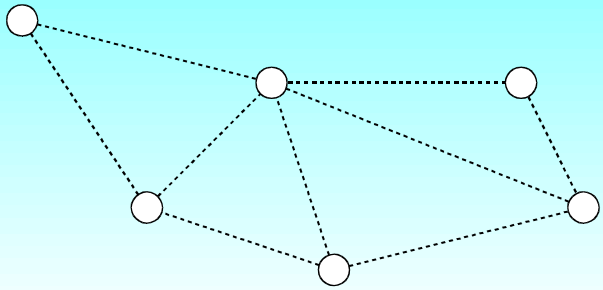
2010

Ten car models have been tested with such a relay hardware. All of them opened and started without problems. The cost of realizing the relay hardware is cheap.

The passive keyless entry and start systems are extremely insecure.

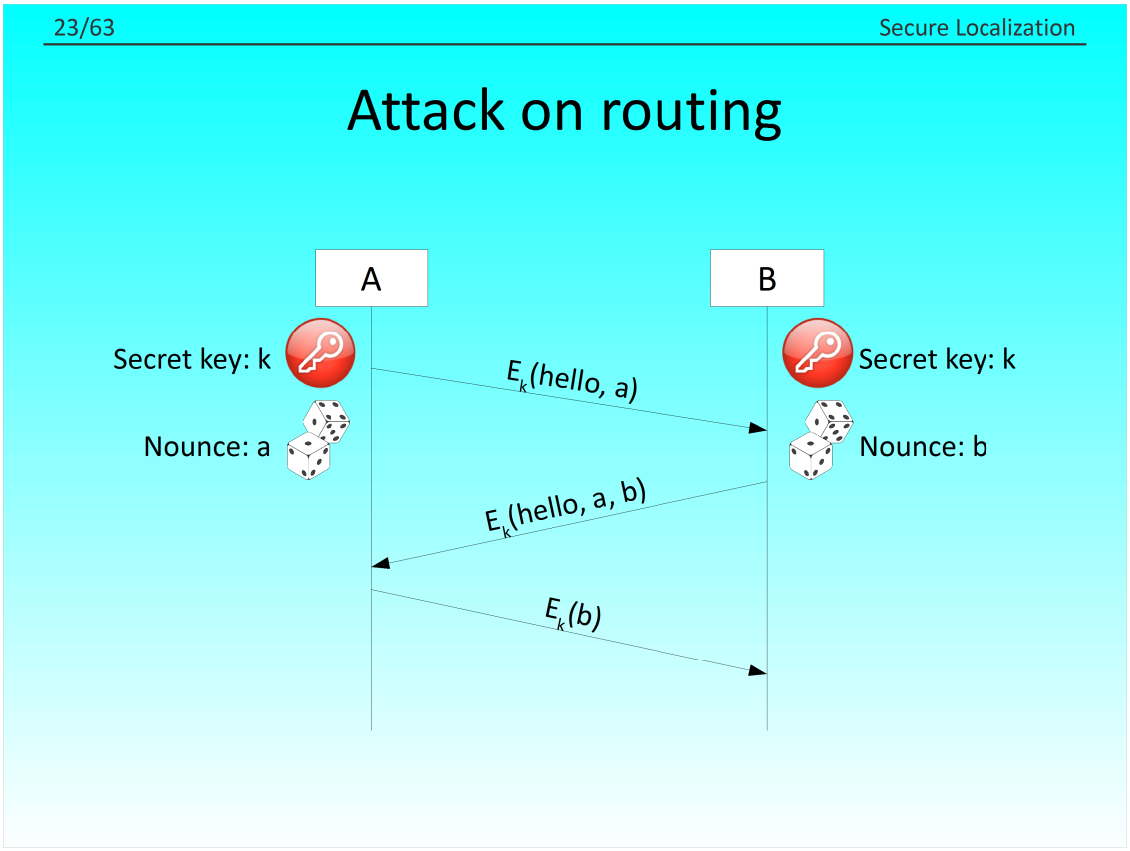
Attack on routing

- An ad-hoc wireless network is critical for a military mission
- It uses an authenticated routing protocol



We can apply the relay principle even in the wireless networks (ad-hoc or sensor networks).

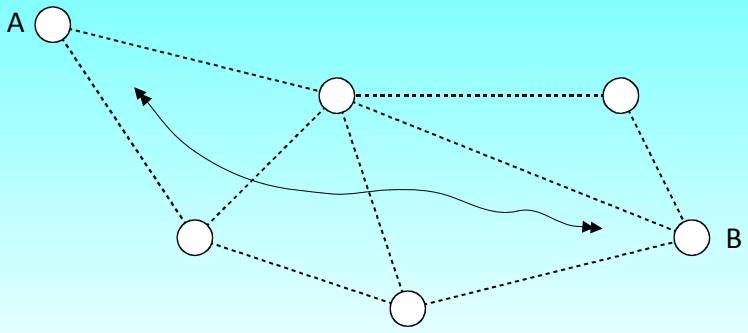
Suppose we have an ad-hoc wireless network for military missions. One of the most fragile mechanisms on wireless networks is routing. Our network uses an authenticated routing protocol.



This is the mechanism for neighbor discovering. It uses classic hello packets and a mutual authentication.

Wormhole attack

- The aim is to create a false link based on a false proximity



The adversary builds a “wormhole”, which is a system of two transceivers. This two devices communicate through a cable, or an RF protocol. The wormhole carries all the hello packets from one end to another and viceversa. The nodes A and B falsely deduce that they are neighbor.

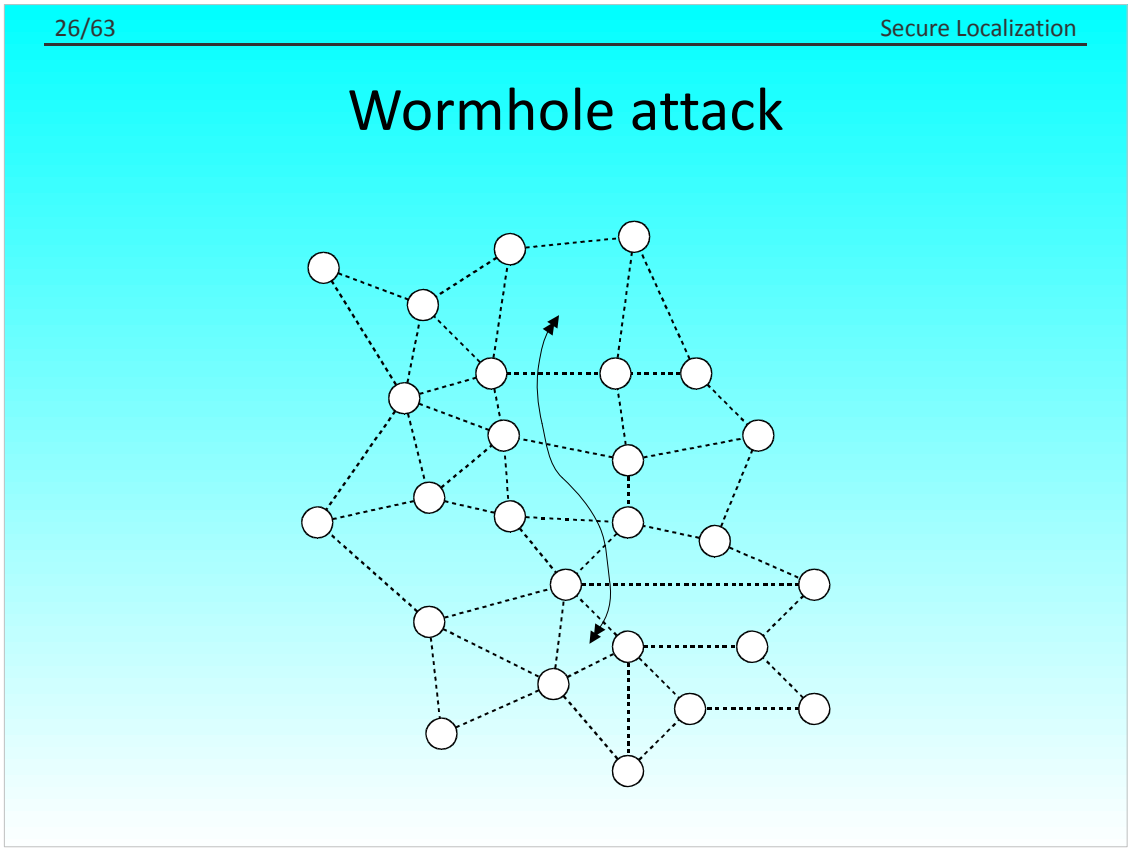
This problem is common for all the wireless routing protocols which rely on “hello” packets to discover neighbors.

Note that A receives route updates also from other (genuine) neighbors, but it discards them. This is because the wormhole link is single-hop, and appears to be very convenient for forwarding.

Wormhole attack

- The adversary has the control of the link A-B
- She can suppress entirely or selectively the communication between A and B

In a sense, A and B actually are neighbors, because they share a link. The problem is that this link is controlled by the adversary. The adversary can suppress entirely or selectively the packets. For example she may suppress the “alarm packets”, while letting go all the other packets.



The problem exacerbates when the wormhole is longer. In such a situation, the northern nodes will choose the (convenient) wormhole link to send packets southern nodes.

Let us see the countermeasures

Though they seem very different problems, they all regard the secure determination of location.

Countermeasures

- *Relay attacks & Mafia fraud*: securely determine the proximity between the prover and the verifier
- *Wormhole*: secure proximity between neighbors or geographical routing with trusted position
- *Truck stealing*: securely determinate the position of the truck

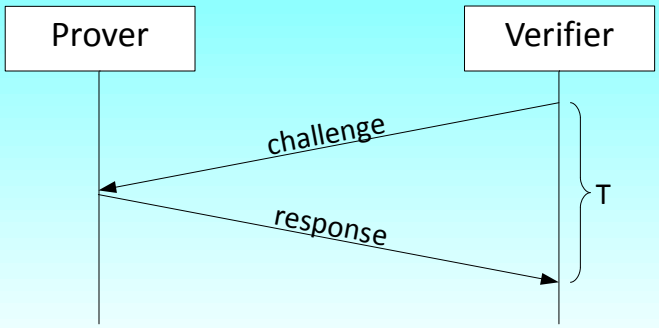
For the relay attacks (access control, PKES) and mafia fraud, we have to securely determine a proximity between the prover and the verifier.

For the wormhole attack, we must either securely determine the proximity between neighbors or use a geographical routing. Geographical routing rely on location information to decide next hops. If such a location information is trusted, the wormhole attack is infeasible.

For truck stealing, we have to securely determine the position of the truck.

Relay attack

- The verifier must be sure about the proximity of the prover
- *Idea:* measure the round-trip time T

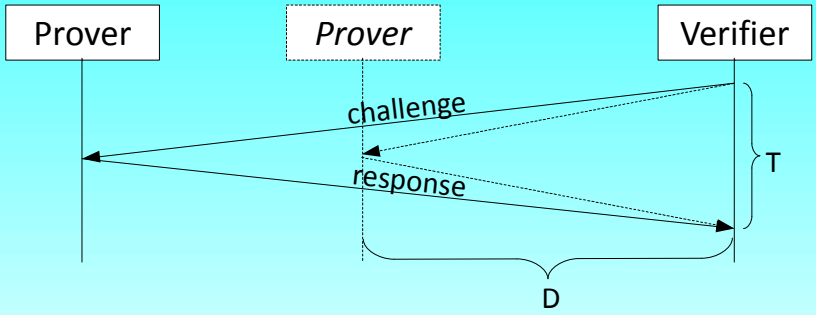


In relay attacks, the verifier must be sure about the proximity of the prover.

The idea is to measure the round-trip time between a challenge and a response. The longer is the round-trip time, the farther will be prover. In order to be sure about the proximity, we limit the maximum round-trip time to a given quantity.

Relay attack

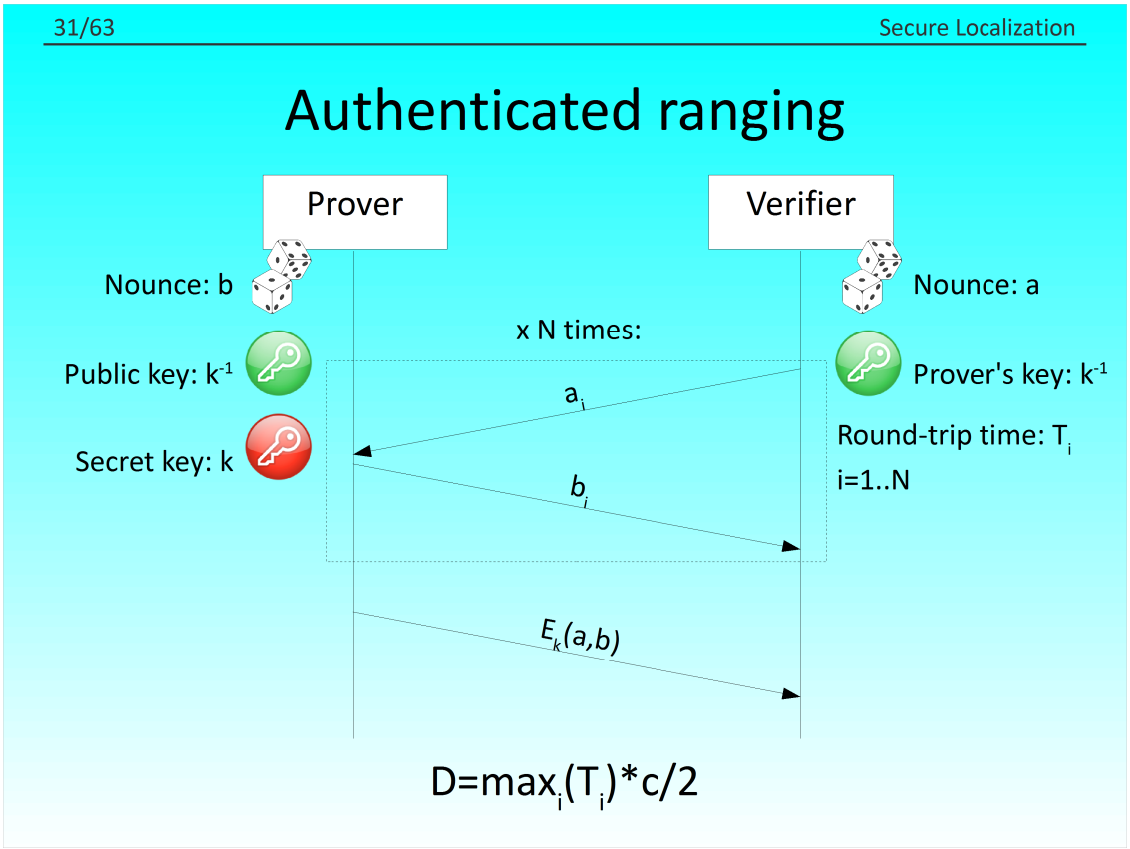
- *Fact:* Who produced the response frame cannot be farther than D=cT/2



- Hence, the prover must be within D meters from the verifier

We are sure that who produced the response frame cannot be more far away than D=cT/2, where c is the speed of light. In fact, if the prover was farther, the challenge and/or the response would have to travel at a super-luminar speed.

Hence, the prover must be within D meters from the verifier.



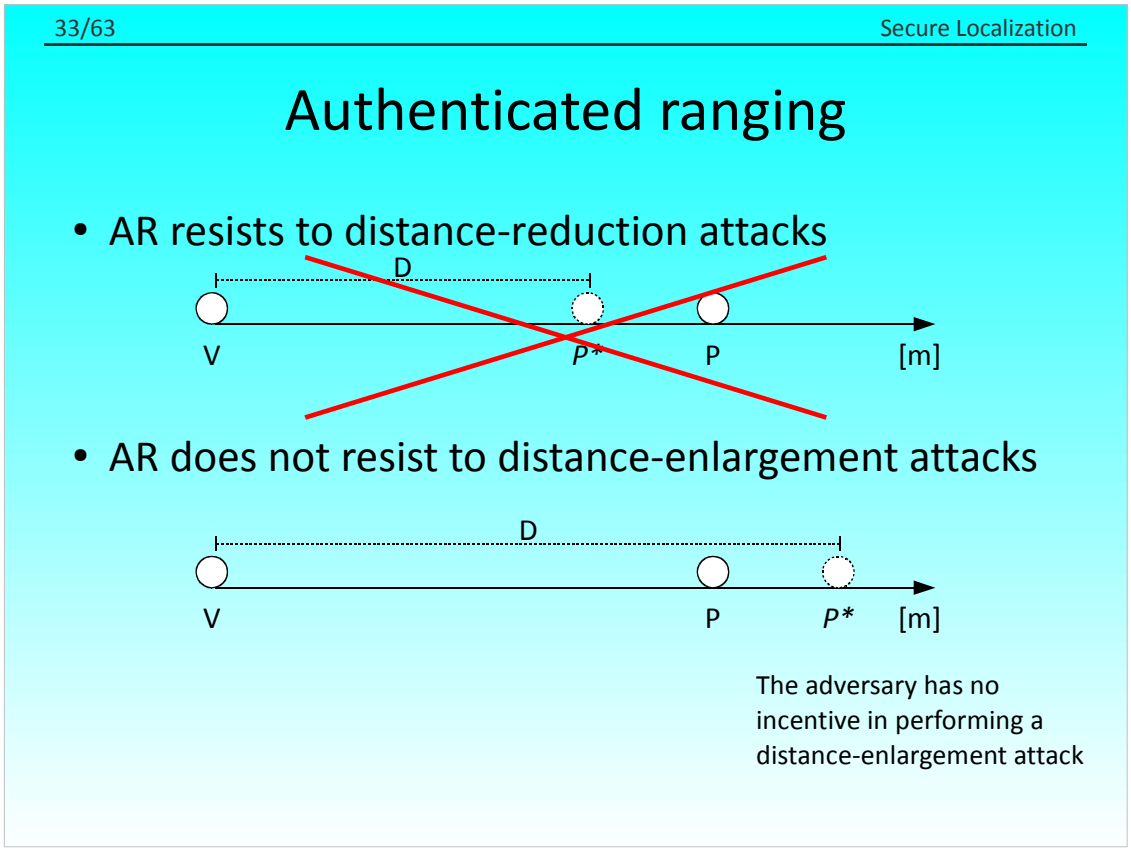
In authenticated ranging protocol, the prover and the verifier determine two random and unpredictable N-bit-long quantities (nounces): a and b. They make N challenge-response rounds. In each round the verifier sends a bit of a and the prover answers with a bit of b. The verifier measures N round-trip times, and determines the maximum distance D=max(Ti)c/2. Finally, the prover signs the nounces with its private key.

A similar scheme is possible with symmetric cryptography as well.

Authenticated ranging

- The verifier checks the distance D
- If P is too far (e.g. D>1m) the access is denied

If the verifier detects that P is too far (e.g. D>1m) the access is denied.



With such a protocol, an adversary wanting to perform a relay attack, must even make the P-V distance appear shorter. But this is impossible, as shown before. In other words, an adversary cannot perform a distance-reduction attack.

On the other hand, it is possible to perform a distance-enlargement attack, that is, make the distance appear larger. However, this attack is useless for breaking the secure proximity determination. An adversary has no incentives in doing so.

Precision problems

- An error of 1 ms on T is an error of 150 Km on D
- The round-trip time must be measured with *nanosecond precision*
  - For radio-frequency signals, this requires ultra-wide band modulations
- Prover's elaboration time T_e must be very small or very predictable
  - Protocol implementation by hardware

Obviously, there are some problems regarding precision. Determining a distance given a round-trip time measurement is not so easy. An error on 1 ms on T corresponds to an error of 150 Km on D!

The round-trip time must be measured with nanosecond precision, which corresponds to 15cm precision on the distance. For radio-frequency signals, this requires ultra-wide band modulation, which could be missing on some devices.

Moreover, the prover elaboration time, which is the time that the prover takes to react to the challenge, must be very small, or alternatively very predictable. This forces to implement the time-constrained part of the protocol by hardware.

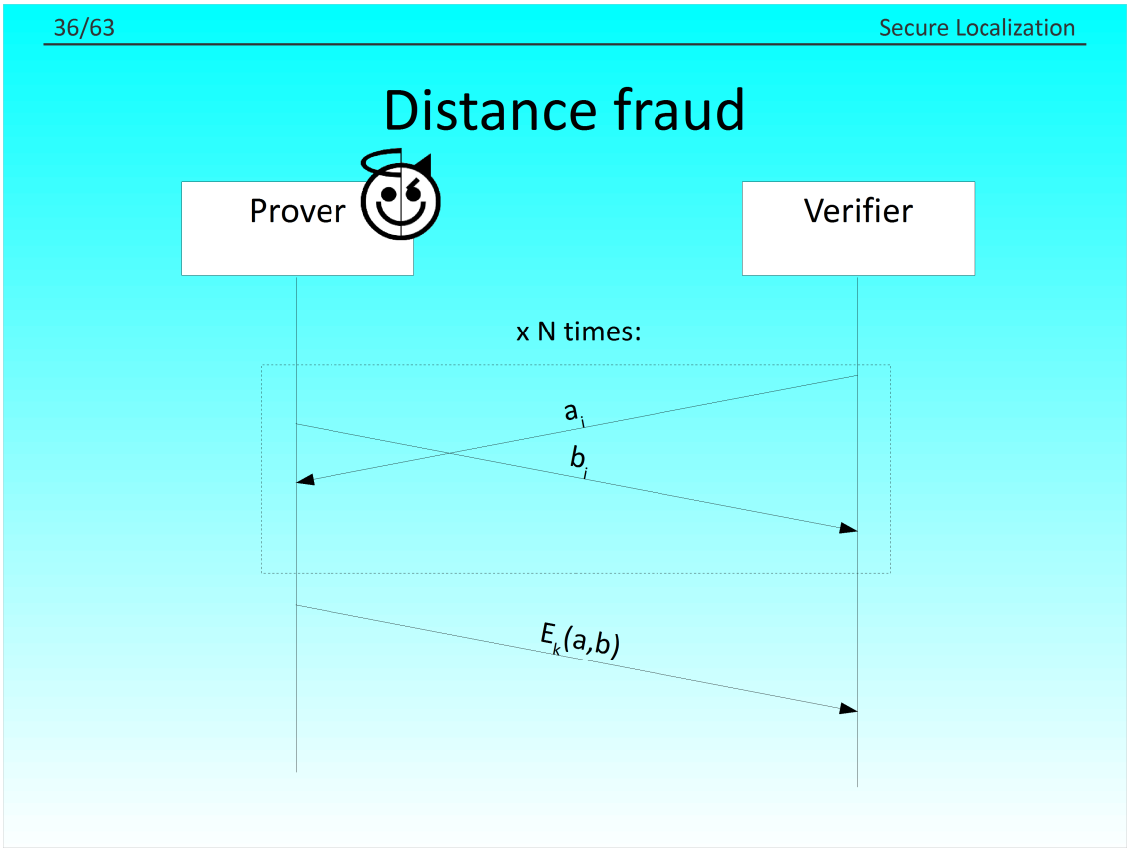
What happens if the prover wants to cheat?

- For example: a company provides its employees with authenticated Internet connection
- Employees can connect via Wi-Fi, but only from inside the office building, not from outside

Until now, we dealt with external adversary, which wants to falsify the distance of a honest prover. What happens if the prover itself is dishonest?

For example, a company provides its employees with authenticated Internet access. They can connect via Wi-Fi, but only if they are inside the office building.

In this scenario the provers (i.e. the employees), have an incentive to cheat about their proximity to the office building.



A dishonest prover could simply send the responses before the arrival of the challenges. This attack is called distance fraud.

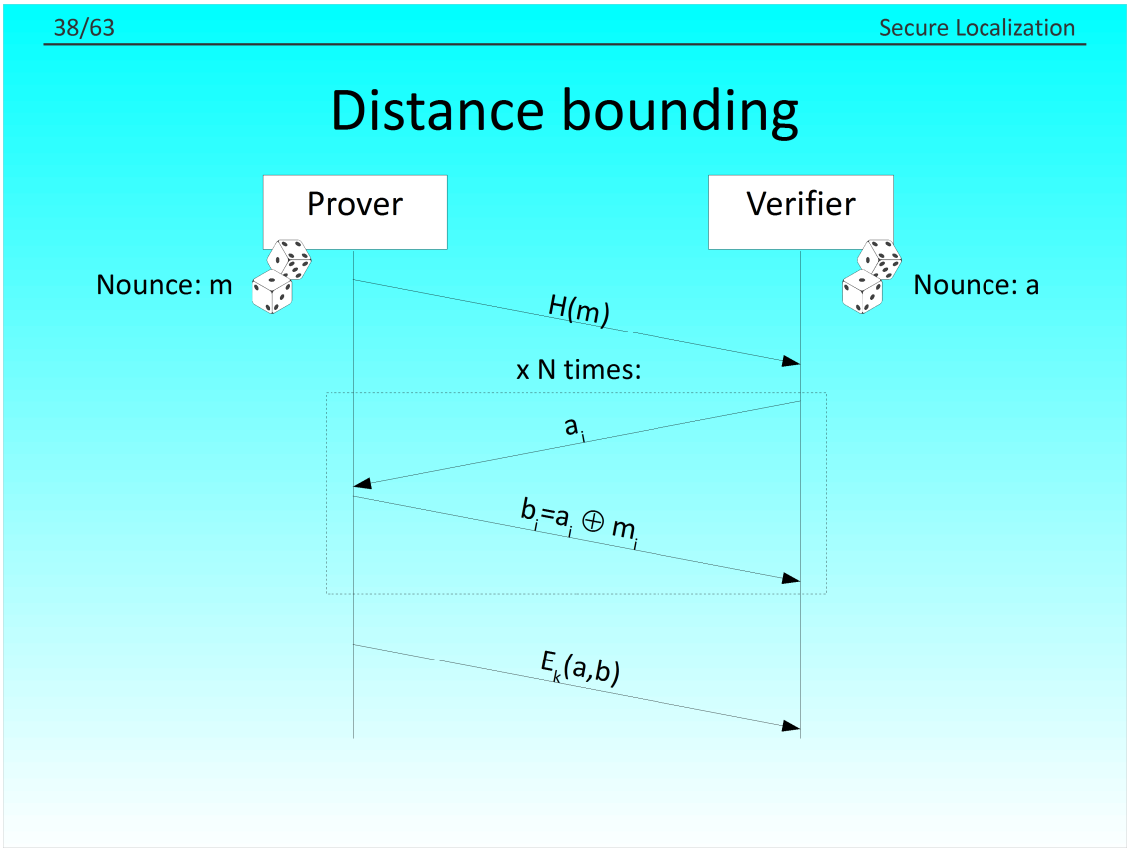
How to withstand distance fraud

- *Idea:* the b_i bits must depend on the a_i bits
- $b = a \oplus m$, with m chosen at random by P

The idea to withstand this attack is simple: the b_i bits must depend on the a_i, in such a way the prover cannot produce them in advance.

Recall that the prover's elaboration time must be small, thus the operation to perform on b bits must be simple. The simplest operation is bit-a-bit exclusive or, with a bit mask m. In order b to be unpredictable also for external adversaries, the prover must choose m at random, and do not reveal it before the challenge-response protocol.

On the other hand, V must be sure that P chosen m before starting

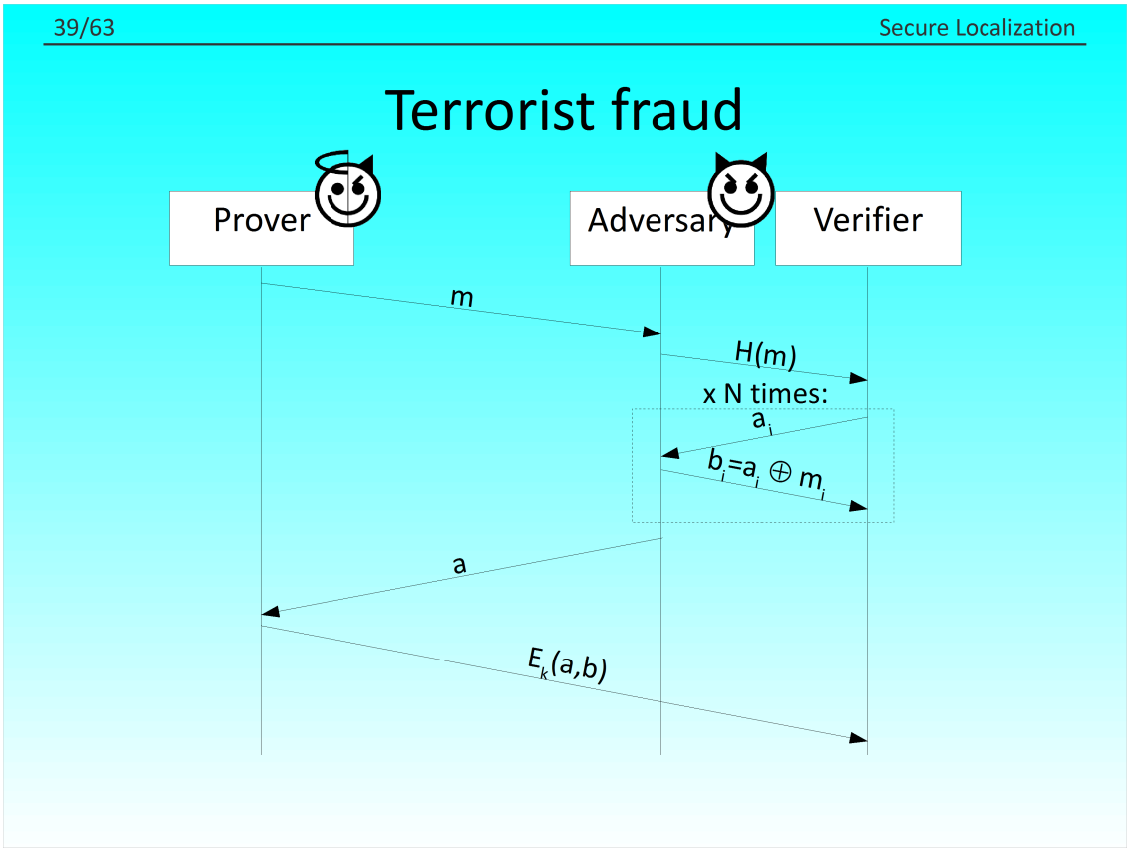


This is the actual distance bounding protocol, which resists both to mafia and distance fraud.

In the commitment phase, the prover determines the quantity m at random and commits to it by sending its hash value to the verifier. The commitment phase assures the verifier that the prover will use a given quantity m, without revealing it.

In the distance bounding phase, prover and verifier perform the rapid bit exchange and the verifier determine the maximum distance D.

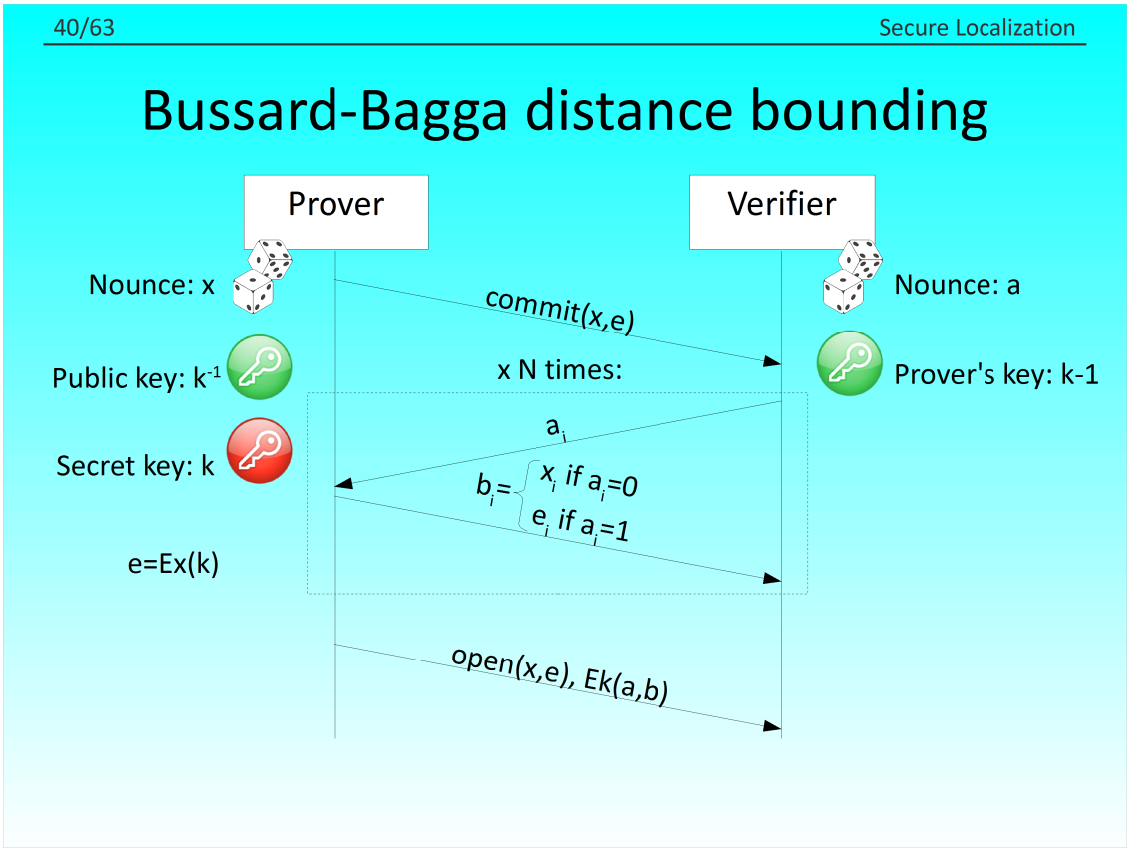
In the proof-of-knowledge phase, the prover signs the quantity a and b with its private key.



We dealt with external adversaries and with dishonestprovers. What happens if these two entities are contemporaneously present? For example, a dishonest employee could let one ally inside the top-secret area, while he is far away.

We suppose that the prover colludes with the external adversary, but he does not want to (or cannot) reveal his long-term secret k. This would be too risky for the prover, because two entities knowing the secret key are indistinguishable from the cryptographic standpoint.

In terrorist fraud, the prover simply reveal to the adversary the quantity m. In this way, the adversary is able to perform the distance bounding phase. Then, the prover signs the protocol.



Bussard and Bagga first resolved the terrorist fraud problem in 2005. The idea is to replace m with two quantities, x and e. x is random and unpredictable, whereas e is the long-term secret ciphered with x.

Only who knows both x and e can perform the distance bounding phase. But who knows x and e can easily compute even k, with a simple decipher operation. Therefore, the prover cannot delegate to his ally the distance bounding phase without compromising the long-term secret.

During the distance-bounding phase, a part of the bits of x and e are compromised, but this does not compromise k.

The commitment schema is more complicated, as the verifier must be sure that the prover used the quantities he committed in advance, without revealing them. Special commitment schemes based on modular arithmetic are used.

Distance bounding implementation

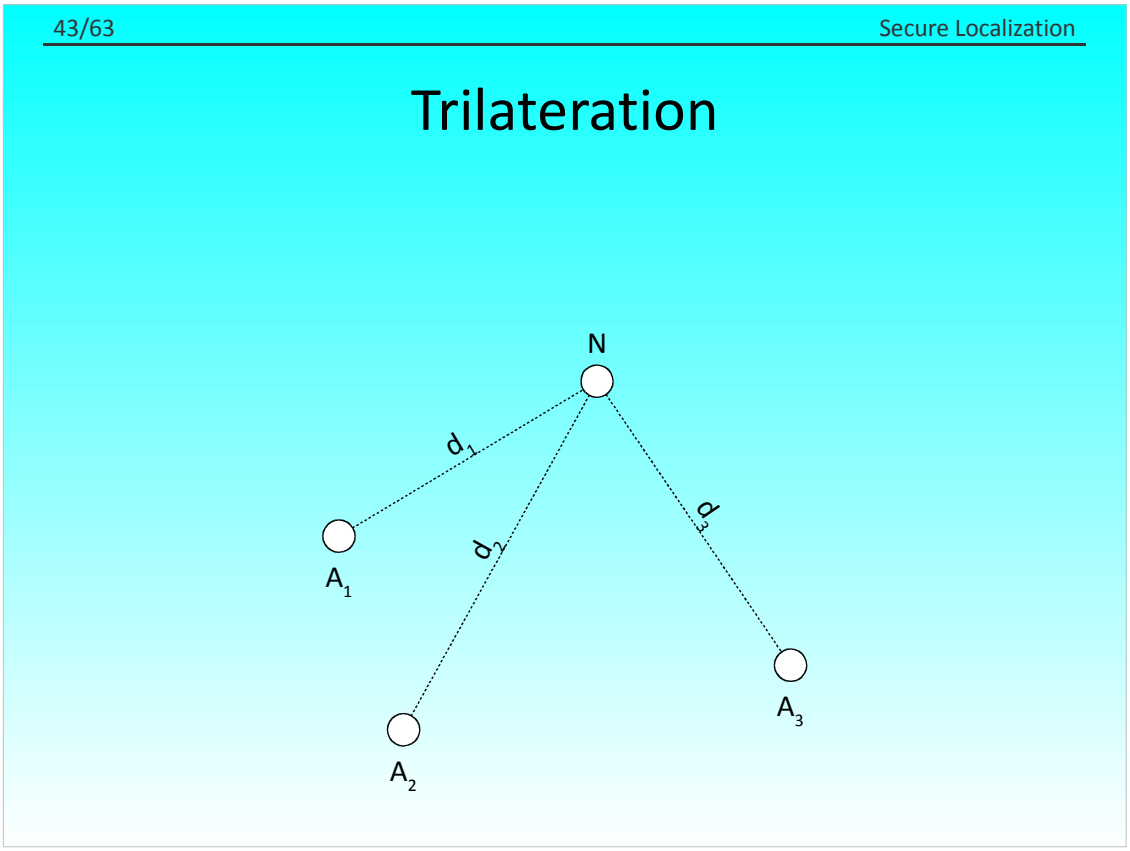
2009

This is an example of distance bounding implementation over radio frequency.

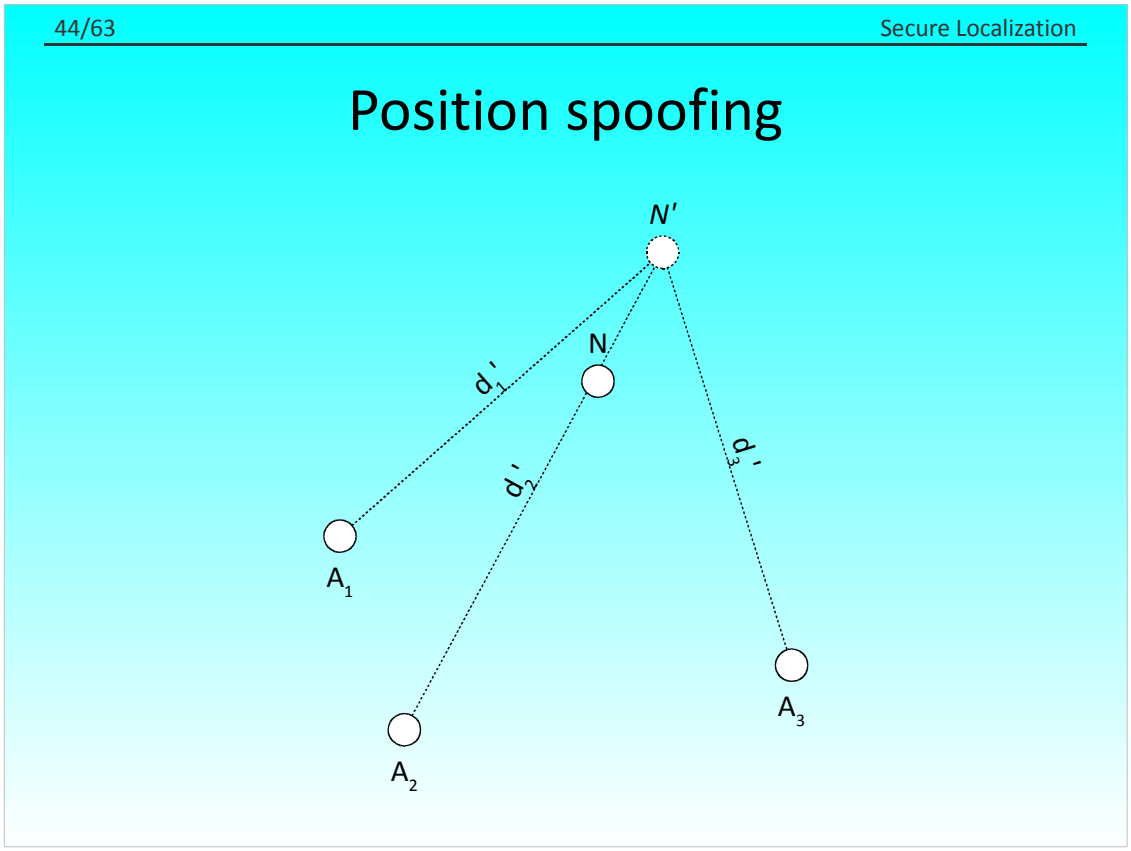
How to withstand truck stealing?

- *Problem:* determining a position in presence of an adversary

Let us now return to the opening problem. How to withstand truck stealing? We must securely determine the position of a device (not its proximity to another device). We cannot use GPS because it is extremely fragile from the security standpoint.

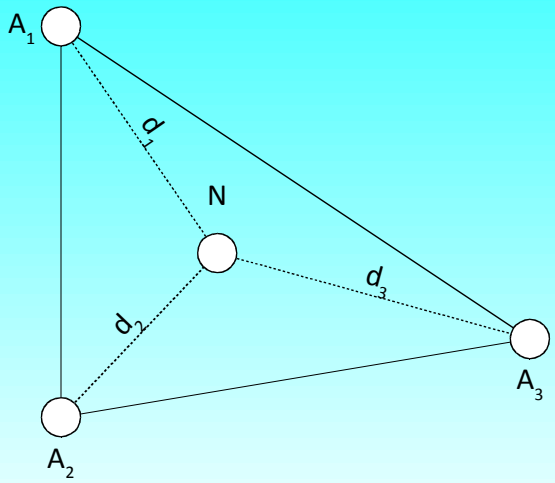


The idea is to trilaterate a position by means of three distance measurements from nodes whose position is known and trusted a priori (anchor nodes). Such distances are determined by the execution of three distance bounding protocols.

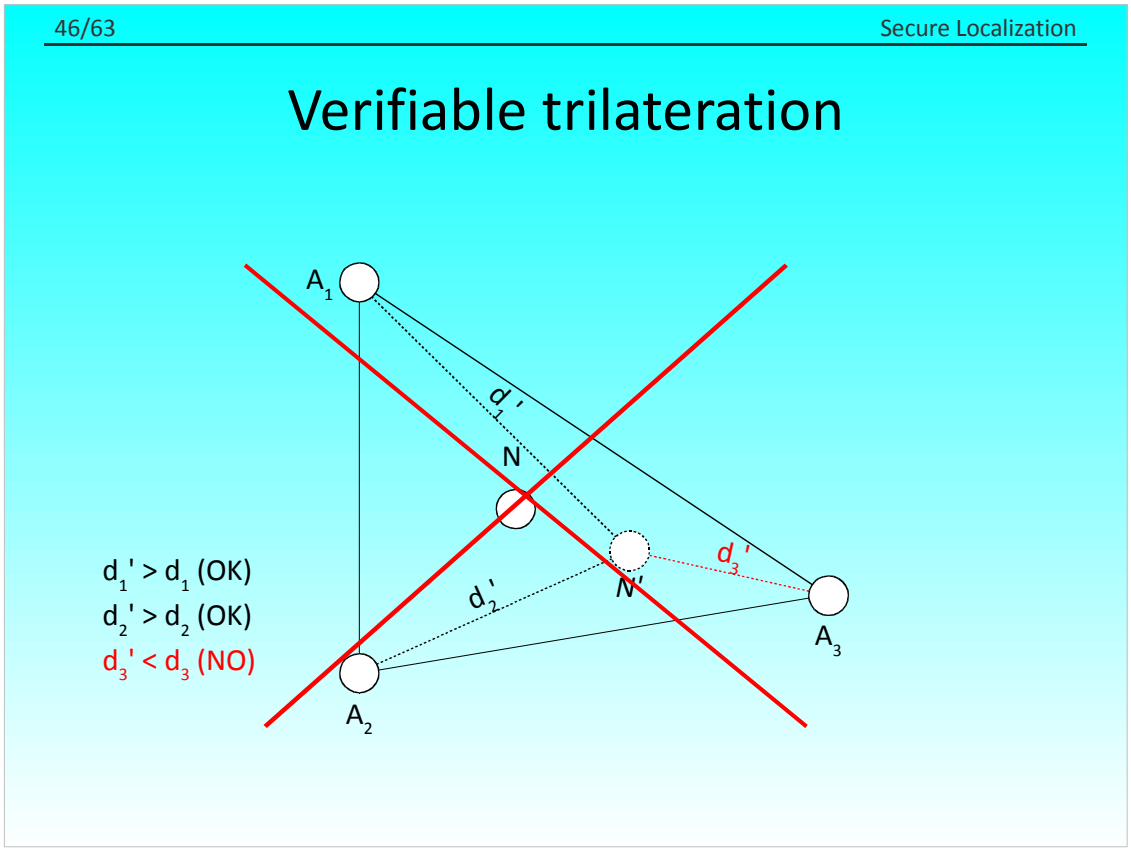


However, the adversary is still able to spoof N's position, by performing three distance enlargement attacks.

Verifiable trilateration

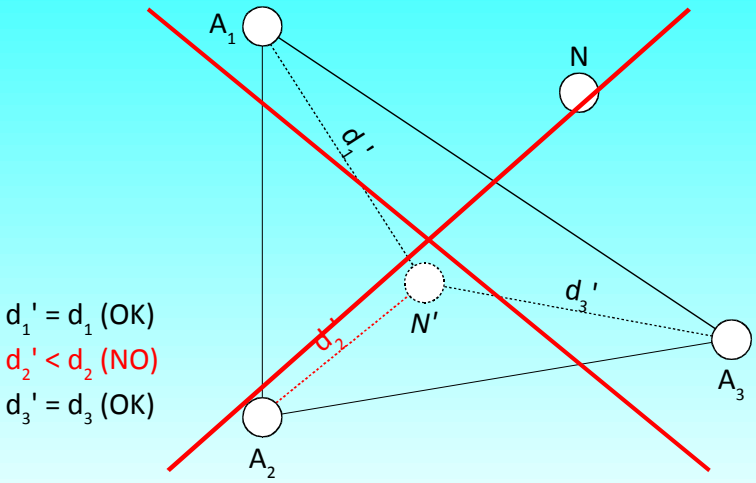


Verifiable trilateration acts this way: First it performs three distance bounding from three anchor nodes, and trilaterates the position of N. Then, it accepts such a position measurement only if it is within the triangle formed by the three anchor nodes. The resulting position is considered trusted.



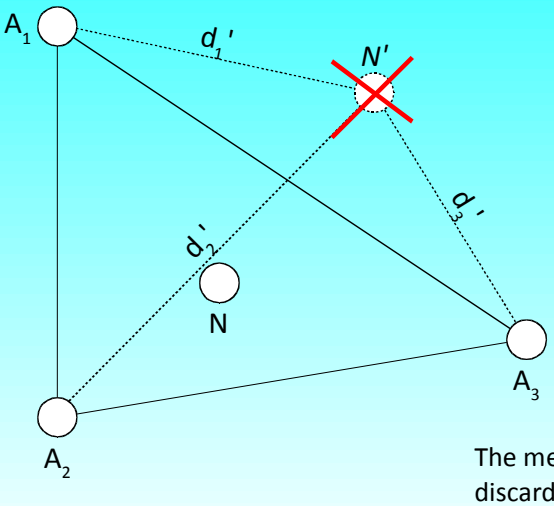
An adversary cannot spoof an internal position to another internal position, because she should perform at least one distance-reduction attack (in this case on d_3). This is infeasible.

Verifiable trilateration



Similarly, an adversary cannot spoof an external position to an internal one.

Verifiable trilateration

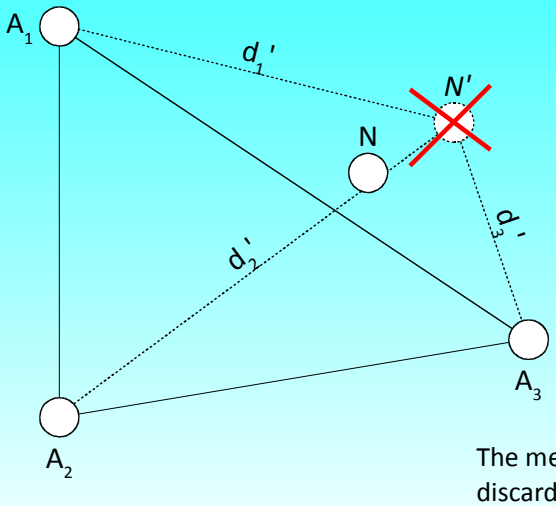


$d_1' = d_1$ (OK)
$d_2' > d_2$ (OK)
$d_3' = d_3$ (OK)

The measurement is discarded.

On the other hand, she can spoof an internal position to an external one, but the system would discard such a measurement. The adversary has no incentives in performing this attack.

Verifiable trilateration

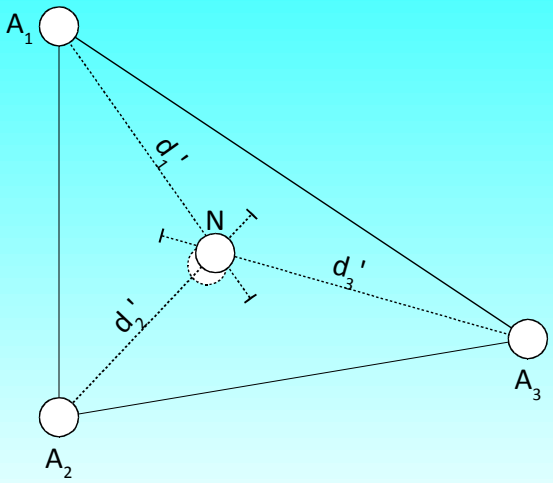


$d_1' > d_1$ (OK)
$d_2' > d_2$ (OK)
$d_3' > d_3$ (OK)

The measurement is discarded.

Similarly, an adversary could spoof an external position to another external position. Again, the system would discard such a measurement. The adversary has no incentives in performing this attack.

Verifiable trilateration



$d_1' > d_1$ (OK)
$d_2' > d_2$ (OK)
$d_3' > d_3$ (OK)

We define a limit precision σ, which discriminates a measurement error from an attack

Finally, the adversary could perform one or more distance-enlargement attacks on an internal position, to make the system measure three “incoherent” distances. That is, three distances which do not intersect on a point.

In such a case, verifiable trilateration behaves like a classic trilateration, and finds the pseudo-solution which minimizes the error squares. As a result, the adversary managed in degrading the precision of the position measurement.

Verifiable trilateration imposes a precision limit σ, which discriminates an ordinary measurement error from an attack.

Verifiable trilateration

- Verifiable trilateration has the same (high) security level of distance bounding
- Distance bounding requires UWB and HW implementations

To sum up, verifiable trilateration offers the same (high) security guarantees of distance bounding. In particular, the probability of success of an adversary is equal to the probability of successfully performing a distance-reduction attack.

However, distance bounding protocols require UWB transceiver and dedicated hardware that could be missing in some devices.

Non-DB methods

- They require less HW resources but offer a lower security level
- The security level depends on the quantity of “good” nodes with respect to “bad” nodes
- They aim at guaranteeing low rates of *false negatives* and *false positives* in attack detection

Other secure positioning methods exist which do not rely on distance bounding.

Their security level is not “absolute” but depends on the quantity of “good” nodes with respect to “bad” nodes. They aim at guaranteeing low rates of false negatives and false positives in attack detection.

Non-DB methods

- SeRLoc (2005)
- ROPE (2005)
- HiRLoc (2006)
- SLS (2006)
- ARMMSE (2008)
- SLAW (2010)
- ...

SeRLoc has been the most influential method in this field. ROPE merges SeRLoc with verifiable trilateration. HiRLoc is an improvement of SeRLoc which gives the same security guarantees but more precision in localization.

SeRLoc

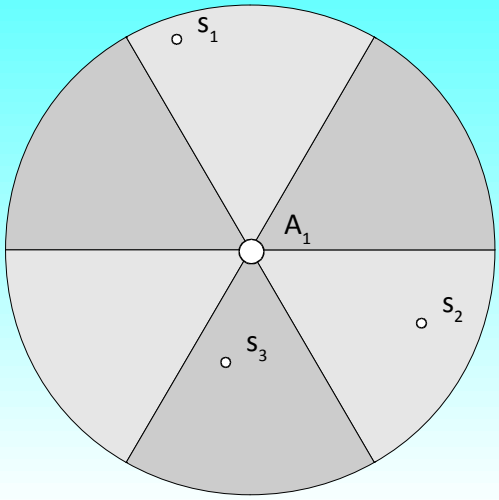
- Secure Range-independent Localization
- Infrastructure of base stations at known and trusted positions: *locators*
- Locators periodically send authenticated beacon packets
- Sensors determine their positions by listening to the beacon packets
- Jamming protection through spread-spectrum modulation and secret spread sequence

SeRLoc means “Secure Range-independent Localization”. It aims specifically at withstanding the wormhole attack. it is range-independent because the localization is performed without distance or angle measurements.

SeRLoc relies on an infrastructure of anchor nodes, called locators. Locators periodically sends beacon packets. The sensors securely determine their positions by listening to the beacon packets. Beacon packets are protected against jamming with a spread-spectrum modulation and a secret spread sequence.

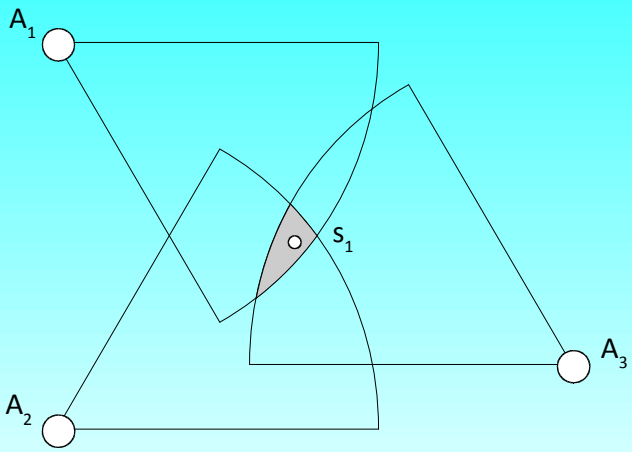
SeRLoc

- Each locator sends contemporaneously N beacons in different range sectors, with directional antennas

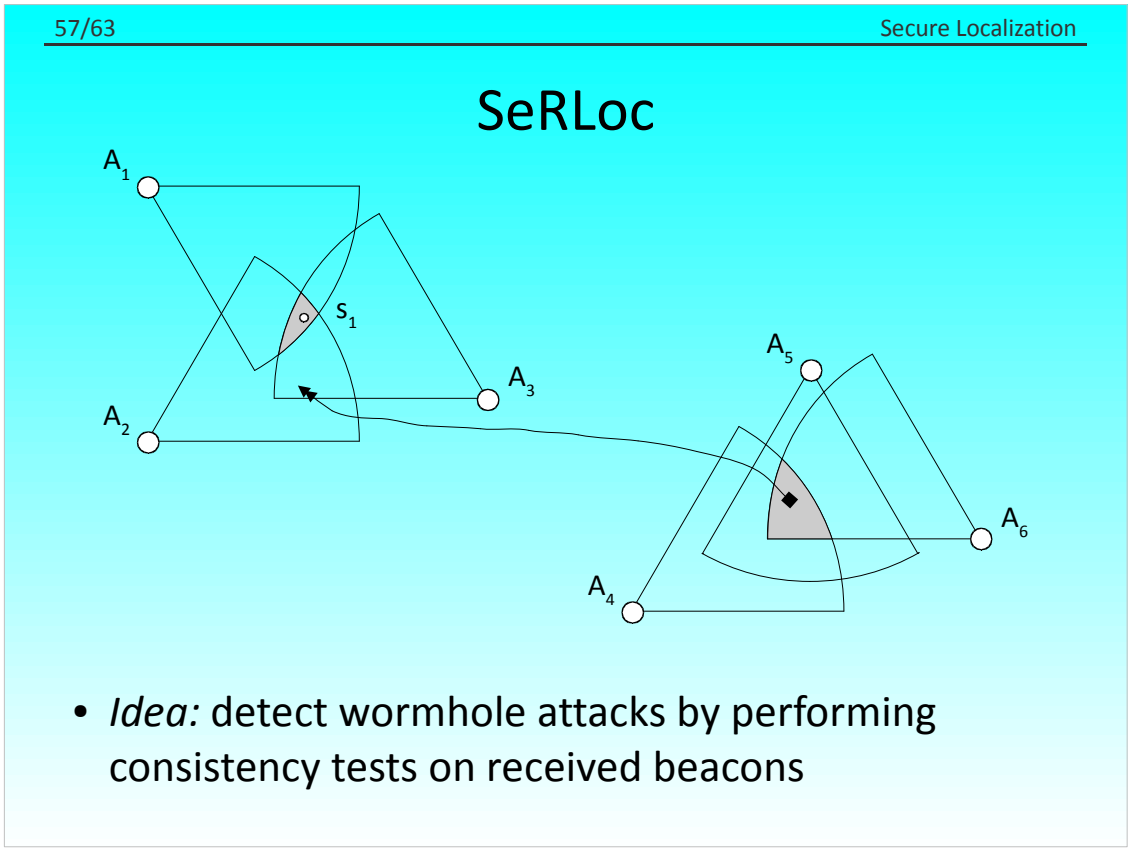


Each locator sends contemporaneously N beacons on different sectors of the transmission area, by means of directional antennas. The beacons are transmitted with a power such as the transmission range is a known quantity R. The beacon packets convey a sequence number, the ID and position of the locator, and the start and end angle of the sector.

SeRLoc



A sensor locates itself passively, by intersecting the sectors of the received beacons.

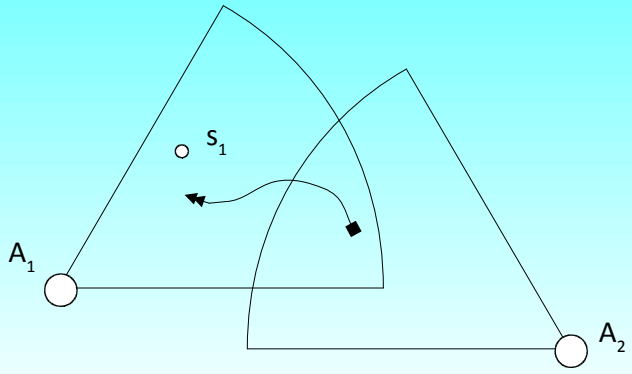


An adversary can try to spoof the position measurement by building unidirectional wormholes. The wormhole replays the beacons received from the origin point to the destination point. The sensor cannot distinguish from directly received beacons and beacons received through the wormhole.

The idea of SeRLoc is to detect wormhole attacks by performing particular consistency tests on the received beacons.

SeRLoc

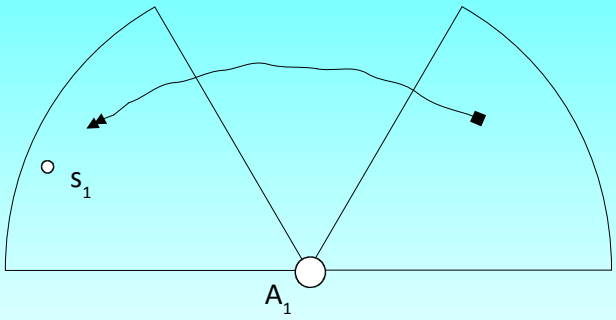
- If a sensor receives twice the same beacon, an attack is detected



For example, if a sensor receives twice the same beacon, an attack is detected.

SeRLoc

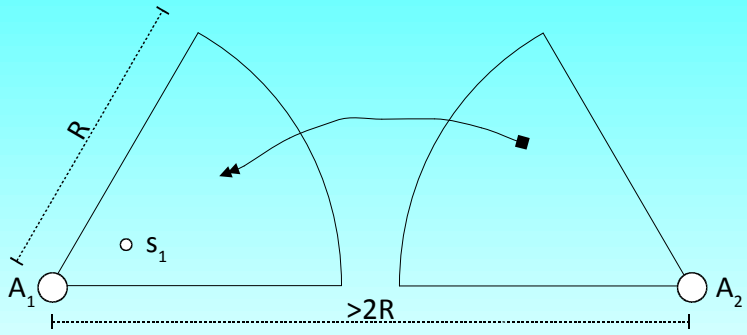
- If a sensor receives two beacons from the same locator, an attack is detected



If a sensor receives two beacons from the same locator, an attack is detected.

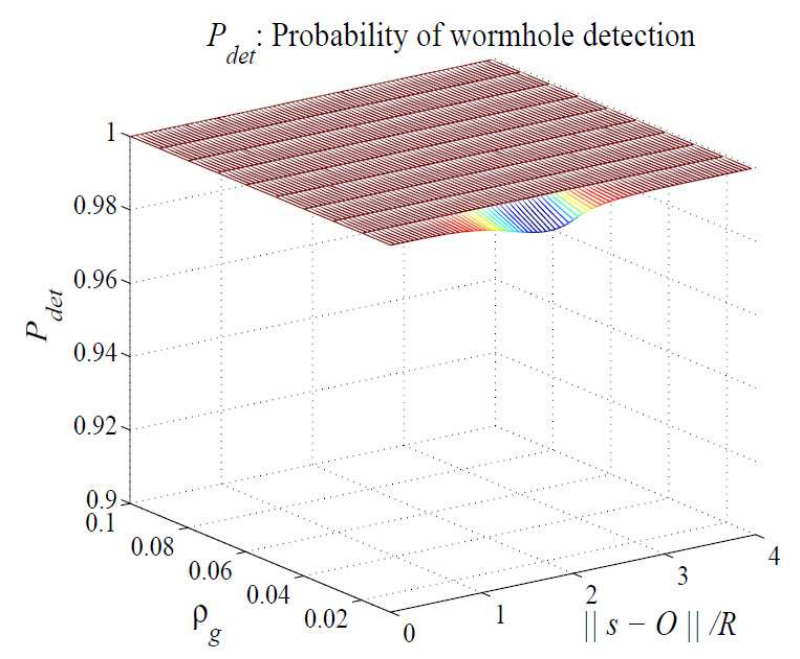
SeRLoc

- If a sensor receives two beacons from locators far from each other, the attack is detected



If a sensor receives two beacons from far away locators, an attack is detected. The locators bust be at a distance greater than 2R, where R is the locator-sensor transmission range.

SeRLoc



The probability of attack detection depends on the locator spatial density and on the ratio between the sensor-wormhole origin and the transmission range R.

Bibliography

- Brands, S., & Chaum, D. (1994). “Distance bounding protocols.”
- Johnston, R. G., & Warner, J. S. (2003). “Think GPS Cargo Tracking = High Security? Think Again.”
- Bussard, L., & Bagga, W. (2005). “Distance-bounding proof of knowledge to avoid real-time attacks.”
- Lazos, L., & Poovendran, R. (2005). “SeRLoc: Robust Localization for Wireless Sensor Networks.”
- Capkun, S., & Hubaux, J.-P. (2006). “Secure Positioning in Wireless Networks.”

Pericle Perazzo

PhD student
Department of Information Engineering
University of Pisa

pericle.perazzo@for.unipi.it
http://www.iet.unipi.it/p.perazzo/