Security of Wireless Networks

{fvisconti}@ethz.ch

ETH Zürich, HS 2022

This is a summary for the course *Security of Wirless Networks HS2022* at ETH Zürich. This summary was originally created during the autumn semester 2020 by thgoebel@ethz.ch. Due to the few changes in syllabus content in the past we have reason to believe that it is also relevant beyond that very semester.

We do not guarantee correctness or completeness, nor is this document endorsed by the lecturers. Feel free to point out any erratas.

Contents

A. Imprint **57**

1. Wireless Basics

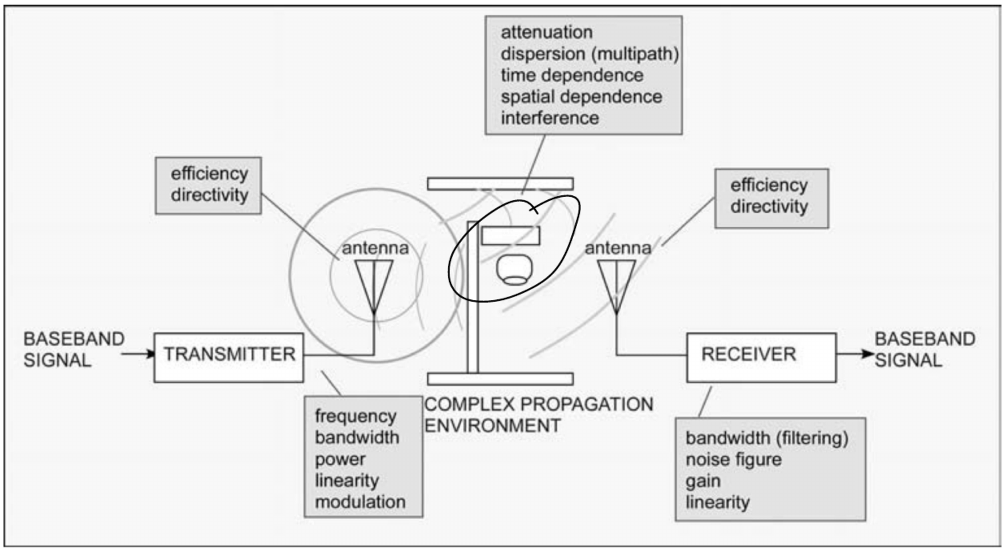

Figure 1: A wireless system, its basic components and characteristic measures

Radio Frequency Signal Electromagnetic radiation, with waves being created in the antenna by an alternating current at the desired frequency. Mathematically described as a function of the time t:

$$v(t) = A\sin(2\pi ft + \phi)$$

with amplitude A, frequency f and phase ϕ. Also recall that the period is $T = \frac{1}{f}$ and the wavelength (distance traveled during one period) is $\lambda = \frac{v}{f}$ (usually $v = c$ speed of light).

Bandwidth The capacity of a communications link to transmit the maximum amount of data from one point to another over a connection in a given amount of time (in bits per second bps). An analogy: The amount of water that can flow through a water pipe.

In other words, the measure of frequency content of the signal. E.g. the human voice contains frequencies in the range from 30 Hz to 10 kHz, and the bandwidth of a single 802.11 channel is 22 MHz.

Often the bandwidth of the base-band and that of the carrier (and thus that of the modulated signal) differ. For example, see spread spectrum techniques (subsection 2.2). Low variability of the signal in time corresponds to a small bandwidth, whereas a high variability corresponds to a large bandwidth

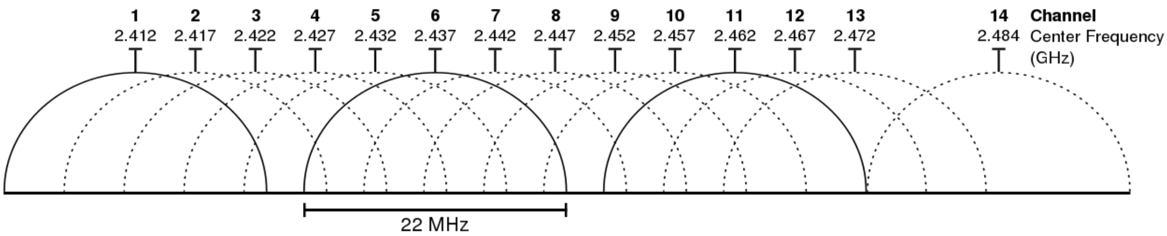

Figure 2: 2.4 GHz WiFi Channels [Source]

Baseband A signal that has not been modulated or has been demodulated to its original frequency content, the actual **information signal**. Telecommunication protocols require base-band signals to be up-converted, or modulated, to a higher frequency in order to be transmitted over long distances.

Carrier A transmitted electromagnetic pulse or wave at a steady base frequency of alternation on which information can be imposed. Typically **a pure sinusoid of a particular frequency and phase** that will carry the information. Usually the frequency of the carrier is much higher than that of the baseband. To go from baseband to passband, we need to multiply the analytic signal with a carrier, where f_c is the carrier frequency and $a(t)$ is the amplitude.

$$x_{RF}(t) = \Re\{a(t)e^{i\theta(t)}\} = a(t)\cos(2\pi f_c t + \theta(t))$$

Upconversion This process is called **up-conversion**, and it's necessary mainly for two reasons: to enable simultaneous transmission of different signals by using a different carrier frequency (for each transmission) and to transform a complex signal into a real one, since only real signals can actually be transmitted. At the receiver, it's then down-converted such that the subsequent processing can be done in the complex-valued baseband domain.

In order to actually perform the up-conversion, we need an oscillator to produce the cosine wave at the chosen (carrier) frequency and a mixer to multiply it with the baseband signal, producing the frequency shift.

Modulated Signal A carrier that has been loaded, or modulated, with the information signal.

Modulation Process of imposing the baseband onto the carrier. The baseband is used to alter one or more aspects of the carrier, such as: signal strength (*amplitude modulation AM*), frequency (*frequency modulation FM*), phase (*phase modulation PM*). In other words, one or more of the values A, f, ϕ in the above equation of the signal are manipulated.

Amplitude-shift keying ASK Modulation technique varying the amplitude of the signal.

Frequency-shift keying FSK Modulation technique varying the frequency of the carrier.

Phase-shift keying PSK Modulation technique varying the phase of the carrier. It's used, for example, in WiFi, RFID, Bluetooth. Specific versions include Binary PSK, Quadrature PSK and Differential PSK. Example: if the baseband bit is 0 do nothing to the carrier, if it is 1 shift the carrier phase by π.

On-Off-Keying OOK Simple form of amplitude-shift keying ASK. Represents data as the presence (1) or absence (0) of a signal. E.g. Morse code.

I-Q Signal Representation A pair of periodic signals are said to be in 'quadrature' when they differ in phase by 90 degrees (e.g. the sine and cosine wave). The 'in-phase' or reference signal is referred to as 'I' (conventionally cosine), and the signal that is shifted by 90 degrees (in quadrature) is called 'Q' (conventionally sine). It's used to represent modulations.

$$I(t) = a(t)\cos(\phi(t)) \leftarrow \text{Analytic signal, } a(t) \text{ is the amplitude}$$

$$Q(t) = a(t)\sin(\phi(t))$$

Antenna Interface between radio waves in the air and electric alternating currents in a conductor. Types include: omni/dipole, yagi, horn, can-tenna.

The directionality of an antenna described how well it transmits/receives into a particular direction.

- **isotropic** — Theoretical, radiates with the same intensity equally in all directions. Often used as a reference antenna when calculating the gain.
- **omnidirectional** — Radiates equally well in all directions in a flat horizontal plane. Most common types in consumer devices.
- **directional** — Radiates best in a given direction by focussing its power. Can thus work with weaker signals than an omnidirectional antenna of the same power.

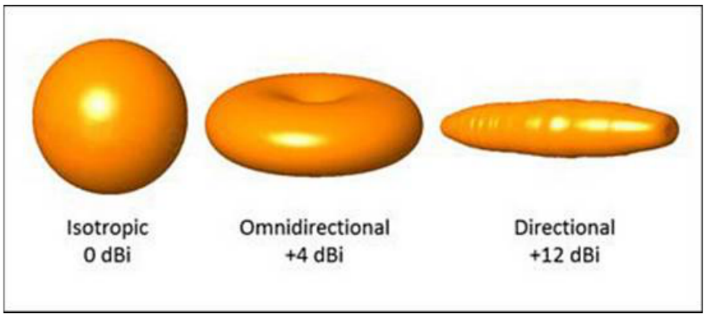

Figure 3: Antenna directionality

Phased Array Array of fixed antennas where the phase of each signal is dynamically adjusted so that the signal will be in phase when viewed from a given direction. Allows *beam steering* towards a specific direction. Possible applications? Can it be used to achieve security (e.g. confidentiality)?

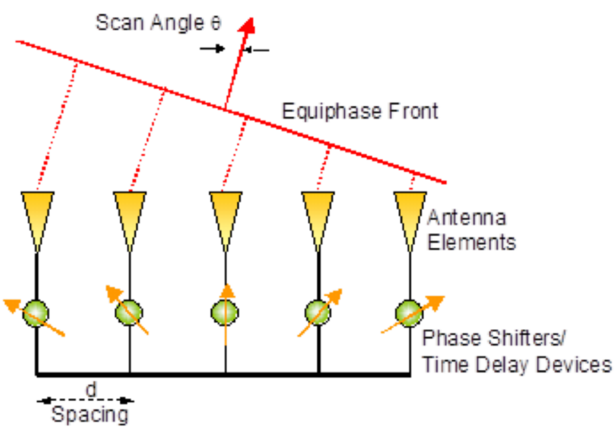

Figure 4: Beam steering

Transmitter/Receiver They convert the signal from digital to analogue or viceversa, apply de-/modulation and connects to the antenna. Properties: transmitted power, carrier frequency, information bandwidth, modulation type, receiver sensitivity.

Software Defined Radio SDR Flexible, low-cost transmitter/receiver. Implements components (mixer, amplifier, de-/modulator) in software rather than processing the signal in hardware.

Channel equation signal strength at the receiver = transm. power + transm. antenna gain − link loss + receiv. antenna gain

See Figure 5. Note that, in free space, the power density of an EM wave obeys the inverse-square law:

$$p \propto \frac{1}{d^2}$$

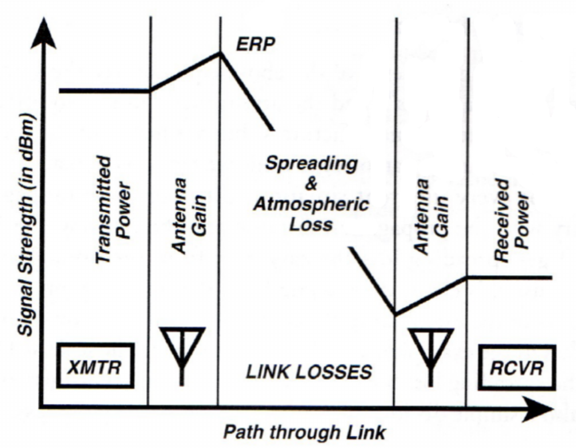

Figure 5: Signal strength across the channel (ERP = Effective Radiated Power)

Receiver sensitivity The weakest signal from which the receiver can still obtain the desired information signal. Depends not just on the antenna gain, but also on other factors such as the noise.

Decibel

- dBm — signal strength in dB / 1 milliwatt mW
- dBW — signal strength in dB / 1 watt W
- dBi — antenna gain in dB / antenna gain of isotopic antenna in dB

Calculating a value in dB

$$dB(n) = 10\log_{10}(n) \quad \text{and} \quad dBm(n) = 10\log_{10}(n/1mW)$$

Power Spectral Density diagram Depicts the power density (in dB) for a range of frequencies. In simple terms, it shows how strong the signal is at a given frequency.

For a signal $x_T(t)$ defined between $(-\frac{T}{2}, \frac{T}{2})$, its power in the time domain will be

$$P = \lim_{T\to\infty} \frac{1}{T} \int |x_T(t)|^2 dt$$

and its power in the frequency domain will be

$$PSD(f) = \lim_{T\to\infty} \frac{1}{T} \int |X_T(f)|^2 df$$

Security Goals Reasons: *security* (integrity, confidentiality, authentication), *regulatory* (personal liability for misuse of one's network access), *safety* (RF-enabled implants).

Just reducing transmission power, hoping that the attacker will be too far away to listen on / send / modify messages, is NOT a solution. In fact, WiFi signals can be received 10 km away, and similarly Bluetooth at 1 km distance (with good, directed equipment).

Example: *Passive Keyless Entry and Start systems (PKES)*, i.e. wireless car keys. Wrongly assume communication implies physical proximity (relay attack). Needs: Authenticated proximity verification, message authentication.

1.1. Questions

Does a wireless channel, in general, affect all frequencies of a signal equally? It depends on what kind of noise/interference is present on the channel. If it's general thermal noise, then yes, all frequencies will be generally affected in a similar way. However, if there are other communications happening, then the frequencies used by those transmissions will have a greater noise level. Also, higher frequency signals suffer from greater attenuation due to the distance traveled compared to lower frequency signals.

Which two properties of the wireless channel are leveraged by systems that rely on those for physical-layer based key establishment? Channel impulse response and received signal strength.

2. Jamming Basics

Jamming Entirely preventing or reducing the ability of communicating parties to pass information, either intentionally or unintentionally.

The jamming signal needs to have the same frequency as the modulated signal. If the latter is unknown to the attacker, they thus need to jam a wide bandwidth of frequencies (containing the band used by the legitimate parties) to be successful.

Effectively, jamming is always a power play.

Symbol Carries one or more bit of information, depending on the modulation scheme.

Symbol Jamming Corrupts symbols such that the receiver can EITHER not interpret them at all OR interprets them incorrectly.
Targeted, low-power jamming of specific symbols is hard!

Communication Jamming Corrupts enough bits that the information cannot be reconstructed any more, despite error correction.

Jamming-to-Signal Ratio J/S $= J - S$, i.e. the difference between the jamming signal and the modulated signal in dB. Usually, a ratio $J/S = 0$ results in successful jamming.

Burn-through range Range in which communication still succeeds, despite jamming.

Attacker model Types: responsive, sweep, random

Actions: jam, insert, modify (= overshadow)

Power to jam/insert/modify: P_j, P_t, P_o

Number of channels to jam/insert/modify: c_j, c_t, c_o

Total strength/power P_T

$$c_j P_j + c_t P_t + c_o P_o \leq P_T$$

2.1. Communication Jamming — LTE

Knowledge about the protocol can allow for more efficient jamming. In LTE, connection establishment relies on information that is transmitted on control channels. Targeting these channels with a (protocol-aware) jamming attack, can allow a DoS without wasting a lot of power using the *Capture Effect*.

2.2. Jamming Resistant Communication

Basic principle If you cannot fight (i.e. have too little power), RUN, HIDE or WAIT, and get ad advantage over the attacker: use a shared secret.

Frequency Hopping Spread Spectrum FHSS Regularly change transmission frequency. The pseudo-random frequency sequence is derived from a shared secret. Sender and receiver **must** be synchronized.

Note that frequency hoppers can be detected and located, simply by looking over time from which direction someone is sending on changing frequencies.

Possible attacks:

- **Partial band jammer:** Distribute jamming power over a subset of all hopping frequencies to achieve $J/S = 0$ at least on that range.
- **Follower jammer:** Detects on which frequency communication occurs and then jams it. Can be protected against by using error codes (since only the final bits will be corrupted).

2.3. Direct Sequence Spread Spectrum DSSS

This technique allows **spreading the baseband over a larger bandwidth using a shared secret** (narrow-band to broadband).
Since the transmission power remains the same, the power density at any given frequency decreases. Thus, the spread signal can effectively *hide under the noise* (Figure 6).

To spread over more frequencies, we need a higher symbol/bit rate. To achieve this, the information signal is multiplied with high-frequency pseudorandom sequences called **chips** or **spreading codes**. The result resembles **white noise**. See Figure 7.

During de-spreading, the signal is again multiplied with the same spreading code. De-spreading converts the wide-band signal into a narrow-band one (this works due to the autocorrelation properties of the spreading code). At the same time, any narrow-band interference is spread out.
Thus, DSSS is more robust against [un]intentional interference and multi-path effects. Broadband jamming is possible, but inherently requires much power.

Detecting DSSS signals is difficult, but not impossible (energy detection of strong signals, signal characteristics such as constant chip rate). Thus, interception and modification is hard.

Example usages: GPS, 802.11b WiFi, CDMA (used in 3G). Non-military applications mainly use DSSS for interference-resistance and use public spreading codes. They are thus still vulnerable to malicious jamming, to cause a DoS.

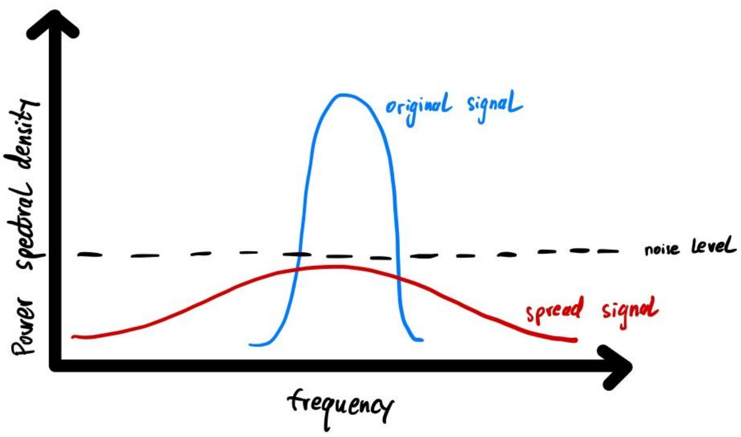

Figure 6: DSSS — hiding under the noise

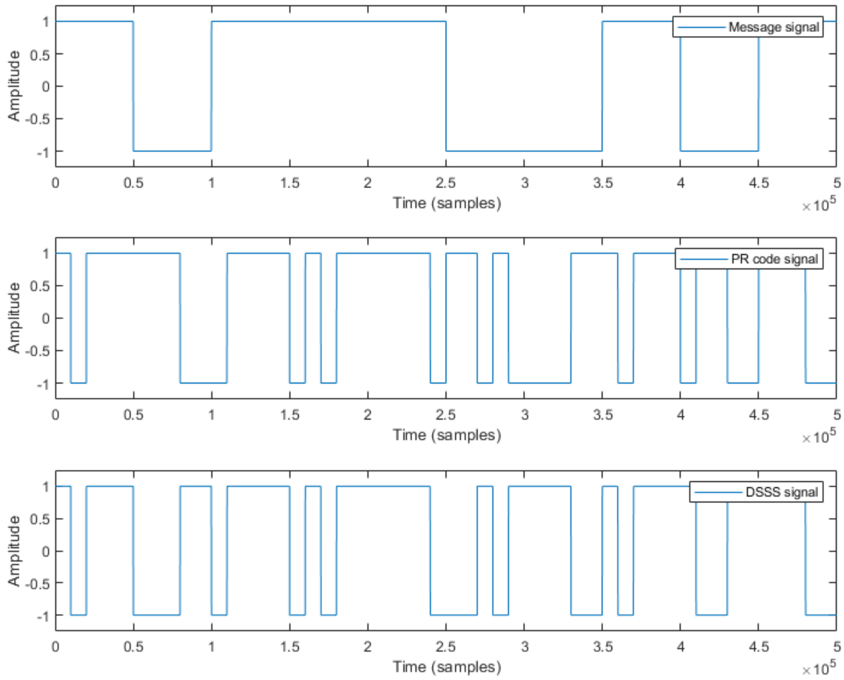

Figure 7: DSSS — baseband signal, spreading code, spread signal (top to bottom)

Processing Gain PG Ratio of the spread bandwidth to the baseband bandwidth, in dB, or chip-rate over symbol rate.

Chirp Signal / Sweep Signal Signal in which the frequency increases and decreases over time ("sweeping" over a bandwidth much wider than the baseband bandwidth). Narrow-band and partial-band jamming are prevented, follower jamming not so much

Code-Division Multiple Access CDMA Multiple transmitters sending in the same area simultaneously, but using different spreading codes. Allows sharing of the same frequencies/bandwidth without interference.

3. Jamming-Resistant Broadcast

Broadcast Communication One sender, many receivers. Inherently open: receivers may join and leave at any time. All receivers listen (c.f. multicast). E.g. radio (FM/AM), GPS.

Challenges when securing broadcast many and unknown receivers, colluding receivers, internal + external attackers. In particular, plain spreading techniques (with group keys) do not work — an internal attacker can use their knowledge to jam other receivers.

Based on FHSS Broadcast Anti-Jamming System due to Desmedt et al.
Base station transmits on multiple frequencies simultaneously. Each receiver listens on a subset of frequencies at a given time. Protects against $j-1$ colluding receivers, ensuring that each receiver has at least one non-jammed channel.

Assumption: the attacker cannot guess the next-hop or detect-and-jam

- **[Public] Channel Allocation Table:** Defines which channels any receiver should listen on, such that $j-1$ receivers do not cover all channels of any other receiver (set coverage).
- **[Secret] Frequency Allocation Table:** Mapping from channel id to frequencies. Derived using a Pseudo-random generator. The complete table is only known to the base station.

Disadvantages: effectively a multicast solution since it requires a shared secret between the base station and each receiver.

Based on DSSS Spreading code is produced by a spreading code generator. Some systems operate with public spreading codes (to mitigate interference). For anti-jamming purposes, pseudo random sequences need to be long and infrequently repeat (wide spread) and they need to have good auto and cross correlation properties.

DSSS hides the signal in the noise. Signal detection is now more difficult (w/o code). Can be done through energy detection(requires strong signal) or signal characteristics (constant chip rate). Signal interception/modification difficult.Narrowband jamming now requires much higher power. Broadband jamming still effective (if you have enough power).

Anti-Jamming—Key-Establishment Dependency Above techniques lead to a circular dependency. We need techniques without shared secrets! Idea: if we cannot coordinate sender and receiver, then don't even try.

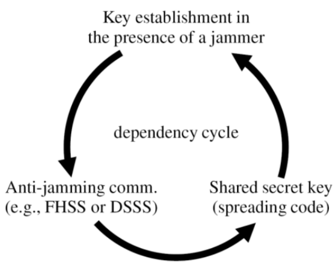

Figure 8: Circular dependency between anti-jamming and key establishment

In addition, pre-loading shared keys is full of problems: requires a trusted party, key revocation, new clients joining, etc.

Uncoordinated Frequency Hopping Spread Spectrum UFHSS Neither attacker nor legitimate receivers can predict which channels are used. Equivalent to FH in terms of jamming protection (but not in throughput).

Transmitter steps:

1. **Fragment** message
2. **Link** fragments (against insertion)
3. **Encode packets** (ECC against jamming)
4. **Repeated transmission** while hopping on frequencies

Receiver steps: same process but reversed (plus packet ordering). Hops from one frequency to the other (sequentially is fine), in the hope of receiving a fragment.

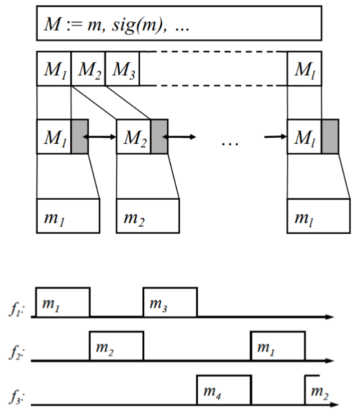

Figure 9: UFH transmitter steps

Issue with fragment linking: the signature is only verified at the end for the entire message.[1] Since there are exponentially many combinations for re-assembly, the attacker can now perform a DoS on a logical (rather than physical) level (pollution attack).
Solution: cryptographic linking of fragments (but without a shared key). E.g. hash linking, one-way accumulators, short signatures.

Disadvantages: Throughput up to 1000x less than Frequency Hopping. Higher latency (depending on attacker strengths, i.e. how high the chances are that the receiver gets a packet).

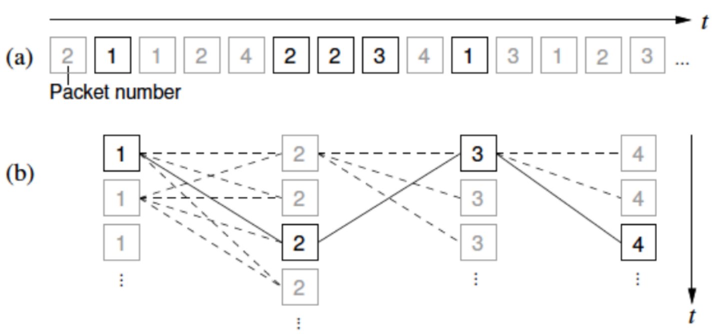

Figure 10: UFH fragment linking — exponentially many candidate messages

[1]The signature is based on public-keys and a mutually trusted — but potentially offline — certificate authority CA.

Uncoordinates Direct Sequence Spread Spectrum UDSSS Neither attacker nor legitimate receivers can predict which spreading codes are used. The public code set C is composed of n code sequences, each containing l spreading codes.[2] De-spreading is done by trial-and-error: it requires the correct code sequence and correct synchronization (which fragment are we at?). The message is also repeatedly sent because of possible jamming — possibly in parallel to improve throughput.

Optimization: first transmit the message M with a secret spreading code K using DSSS. Then transmit the spreading code K using UDSSS.

Advantage: quicker decoding, longer messages, flexible security level.

[2] This allows a message to be fragmented into l pieces.

4. Security of Global Navigation Satellite Systems — GNSS

Overview Orbiting satellites transmit their location and a precise timestamp. Receivers collect these navigation messages and their arrival time and use **trilateration** to calculate their own position. [3] Satellites are positioned in such a way that at least four of them are always in sight from any point on Earth. If more satellites are visible, a more precise localization is possible.

Remark: even though space is three-dimensional, we need at least 4 satellites (thus 4 equations) because we also need to solve for the clock-error, which is otherwise unknown.

Three segments: users, satellites, ground control.

Types of start at the receiver intuition: the more the receiver knows, the faster it can lock on to the satellites in view and get a new localization.

- **Cold start:** receiver knows nothing.
- **Warm start:** receiver remembers last calculated position, almanac and UTC time.
- **Hot start:** receiver remembers last calculated position, almanac and UTC time and the last satellites in view.
- **+ Assisted GPS:** receiver gets helped by the cellular network downloading ephemeris and estimating its position using triangulation from cell towers.

Signalling Each satellite **modulates the navigation message** with a **spreading code** (coarse acquisition C/A for civilians (public), precision P/Y for military (secret)). The spread signal is then modulated onto a carrier. Individual satellites use individual spreading codes to allow distinction. For the civilian GPS, spreading codes are **public**.

GPS sends on two carrier frequencies at the same time, L1 (1575.42 MHz = 10.23 MHz $\times$ 154) and L2 (1227.60 MHz = 10.23 MHz $\times$ 120).[4] Apart from jamming resistance and redundancy, this also allows to calculate the ionospheric delay error.
Due to atmospheric attenuation, down on Earth the GPS signal is well below the thermal noise.

Navigation message Each message consists of 25 frames. Each frame takes 30 sec to transmit, so the total time, in the case of a cold start, is 12.5 min.
Each frame contains: satellite clock + health data, 2x ephemeris (orbit details), other data + almanac (orbital + clock details). Navigation messages are transmitted by the satellite at a rate of 50 bits per second.

Time of Arrival TOA The time of arrival is the travel time of the signal from the satellite to the receiver and it's used to calculate the distance and, thus, the receiver position. It's found by sliding the spreading code over the received message until a correlation peak is found.

[3] There are of course issues with special and general relativity that mess with the time.
[4] This only applies to military. The civilian C/A is only transmitted on L1.

Spoofing attacks Messages are unauthenticated (for practical reasons, or otherwise they would become too long).

By sending stronger signals, which overshadow the legitimate ones, an attacker can modify the *navigation message contents* (transmission time, satellite location) or their *time of arrival* (retransmitting captured signals with a temporal shift), resulting in a wrong location being calculated.

This is an issue in civilian GPS (messages can be generated and delayed) as well as in military GPS (messages can only be delayed since they are encrypted). Unfortunately, commercial GPS signal generators are becoming increasingly cheap.

Impct of attacks Jamming can cause DoS, which is bad but it's possible to recover in other ways. However, it's useful to assist the attacker in other attacks (for example jamming the legitimate signal before injecting a rogue signal). Obtaining a rogue position is bad for smart/autonomous vehicles, high-value asset tracking, especially if you don't detect it and keep trusting the wrong position.

4.1. Spoofing Detection and Mitigation

Types of countermeasures

- **Infrastructure/protocol:** for example, cryptographic authentication of navigation messages
- **Receivers:** for example, using physical-layer characteristics of the signal to validate the signal as well as the calculated position/velocity/time (e.g. direction of arrival, carrier phase, signal strength, etc...)

Angle of Arrival — AoA Use multiple antennas (e.g. on both ends of a ship) to calculate the angle of arrival through the phase difference and the known distance between the antennas (see beam steering, Figure 4). In a spoofed scenario, the angles would all be very similar. Restricts the locations from which the attacker can successfully spoof.

<u>Problems:</u> Attacker can use drones to spoof signal from more realistic angle. Reflection of legitimate signal of buildings (thus reaching the receiver at a shallower angle) could be wrongly classified as spoofing. Computationally expensive phase measurement. Hardware modification.

Monitor Signal Characteristic Changes Over time, monitor signal properties such as AGC (Automatic Gain Control), noise level, number of satellites, spatial diversity (AoA) or the autocorrelation peak. Abrupt changes in any of these indicate presence of spoofing.

Seamless takeover attack The attacker starts transmitting a copy of a legitimate GPS signal in sync with the original one, but at low power, having no influence on the receiver. Then the attacker slowly starts increasing the power, until the receiver prefers the attacker signal. Now the attacker can change the GPS signal, and the receiver will keep following.

SPoofing REsistance GPS rEceiver SPREE Leverage peak tracking (of <u>all</u> signal peaks) to detect seamless takeover attacks. Navigation message inspection detects content spoofing.

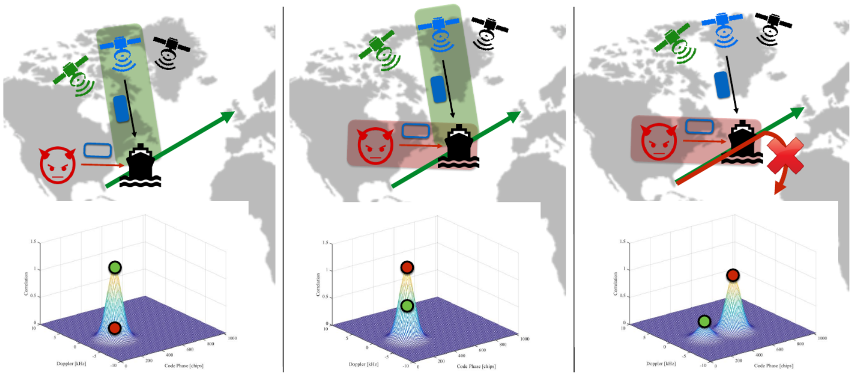

Figure 11: Seamless takeover attack

Cryptographic approach (Kuhn)

1. At time t: satellite uses secret spreading code. Receiver uses a broadband receiver to capture the entire band.
2. At time $t+dt$: Satellite disclosed code, signing the disclosure with its secret key. Receiver verifies signature, de-spreads the signal.

Advantages: Prevents fake signal generation and individual signal delay.

Disadvantages: Requires pre-shared public satellite keys. Slightly inefficient (longer latency until signal lock). Requires loose synchronization (after dt the attacker knows the spreading code and can spoof). Does NOT prevent full-band delay. Relay/replay attacks (record signal and replay full band in another location in real time).

Galileo OSNMA Galileo Open Service Navigation Message authentication offers a way of authenticating messages that's sustainable for the GNSS infrastructure.

Solutions that use public key cryptography are not suitable, since the signatures would be too long to fit in the limited bandwidth available, and are computationally too expensive. Symmetric cryptography is not suitable as well, since it relies on shared secrets that would need to be stored in every receiver, which one can't trust to actually stay secret.

OSNMA uses time-delayed authentication. The satellite generates a random value K_n and hashes it n times and obtains a chain. Obviously, the hash function is irreversible, so you can't compute K_n if you have K_{n-1}. The satellite sends the digital signature and then starts sending the messages in the form of

$$M_1, K_0, MAC(M_1, K_1) \rightarrow \text{ then, after some time, } \rightarrow M_2, K_1, MAC(M_2, K_2).$$

Basically, it discloses the key only after that the message encrypted with that key has been received — and thus the key, which is valid only once, is no longer useful to any attacker. The receiver verifies the first key against the signature using the public key, then verifies the key against the previous key by hashing it and then verifies the previous message using the key disclosed at that moment.

Security analysis It's important to note that this solution is still susceptible to anticipation attacks, which are harder but possible and use techniques like Early Detect-Late Commit and Forward Error Estimation, and also delaying attacks, butthese can be detected with a clock offset test.

5. Physical Layer-based Security

5.1. Broadcast Authentication: Presence Awareness

Scenario One broadcasting station, many receivers. No pre-shared keys, no credentials (certificates, public keys, etc). Receivers know that they are within the power range of the legitimate sender (reasonable in airports, universities, etc). Receivers know which channel (frequency) the sender is broadcasting. The sender is always on and transmitting.

Goal: distinguish between broadcast messages of legitimate and malicious station. In other words, a physical layer-based broadcast authentication scheme based on presence awareness. See also subsection 6.1.

Integrity Codes Protocol

Sender: Spread message m from k to $2k$ bits using Manchester encoding ($1 \rightarrow 10$, $0 \rightarrow 01$). Transmit result using on-off keying.
Receiver: Set power thresholds above which to interpret a signal as 1. Then decode.
Integrity Verification: Check if hamming weight[5] $H(m) = |m|/2$. If yes, the message was not modified.

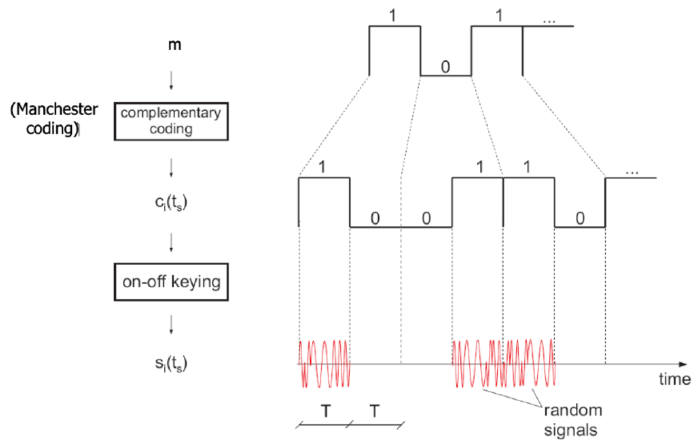

Figure 12: Integrity Codes

Hardness assumption The attacker cannot block the legitimate signal from reaching the receiver antenna. Also, it needs to be hard to annihilate 1 bits, (turning bits with value 1 into bits with value 0 by "guessing" the random signal -different from zero- to encode bit 1) — using randomized symbols, for example.

Analysis

- The attacker can easily change $0 \rightarrow 1$ in the raw/keyed signal. They can change $1 \rightarrow 0$ only with small probability (assuming signal annihilation is hard).[6]
- How to handle arbitrary length messages? Solution: Use i-delimiters between messages. In Manchester encoding e.g. 111000.

[5]Recall that the hamming weight is the number of bits unequal to 0.
[6]thgoebel: Why is this so? Just a couple of slides ahead it was explained that signal annihilation is doable? — you use random symbols for this reason

- Slow. Solution: broadcast hash of message using integrity codes, use another faster channel for the full message.

5.2. Channel-based Key Establishment

Uncertainty of the Wireless Channel In a complex, multi-path rich environment, the wireless channel exhibits time-varying, stochastic and reciprocal fading. For receivers separated by $> \lambda/2$ their channels are not correlated.

Thus the channels between sender S and receiver R are "random" and cannot be known/predicted by the attacker. In particular, the attacker cannot remotely measure multi-path fading components of the signal strength.

Key Agreement through Channel Properties We can leverage different properties of the channel, e.g. Channel Impulse Response, RSSI[7], CIR[8] or signal phase.

Main assumptions: the **attacker is located at least half a wavelength away** from communicating parties; the attacker's channel measurements are de-correlated from those computed by the other parties, therefore likely no have access to the measured secret randomness; **if the attacker injects** signals/interference during the key generation, the attacker signal is measured differently at the benign devices due to channel distortions and the **result is key disagreement**.

The generic steps are the following:

1. Signal Acquisition and Quantization
2. Reconciliation (error correction, privacy amplification)
3. Key confirmation

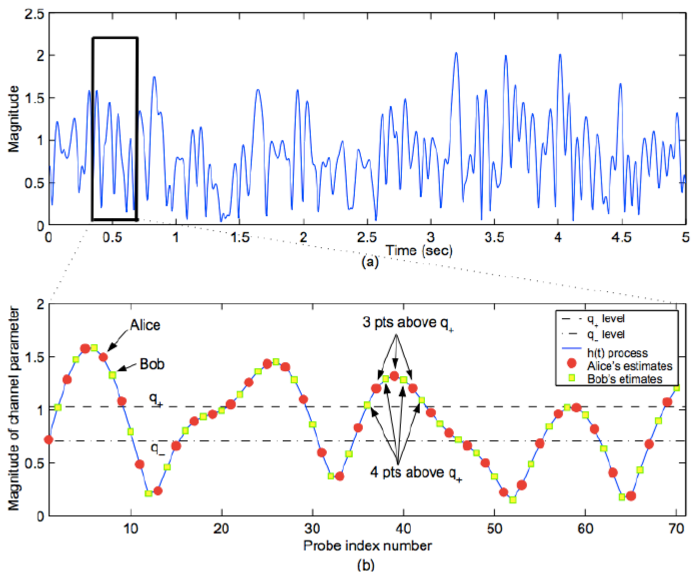

Figure 13: Measuring Channel Property Magnitudes over Time

[7]Received Signal Strength Indication
[8]Carrier-to-interference ratio

Analysis There are several disadvantages to this scheme:

- No authentication: we establish a secret key, but with whom?
- No guarantees on the environment: Multi-path rich? Cannot be pre-measured? Receivers (and attacker!) at least $> \lambda/2$ from each other?
- Questionable benefit over classic public-private-key schemes (no information-theoretic security)
- Active attacker not considered: can influence and discover the key

Ensuring Secrecy with MIMO The goal is to strengthen the security of the previous scheme. Idea: **use multiple antennas on both ends**, to (a) steer the signal towards the receiver and away from the attacker and (b) use jamming to interfere with the attacker (but not the receiver).

Note that we can model each channel (or the signal on that channel) as a complex number (amplitude as the real part, phase as the imaginary part).

Zero Forcing Assumption: S knows the channels to the intended receiver R_1 and to the attackers R_2, R_3, which are given as channel matrices H_r. The sender applies a transmission filter F that is constructed such that $R_r = HFD = HS$ contains useful data for R_1 but not for the attackers.

Orthogonal Blinding Same as zero forcing, but we only assume that the channel to the intended receiver R_1 is known (but not the channels to the attacker). Construct the filter F such that for everybody but R_1 the result R_r contains a jamming signal (i.e. noise).

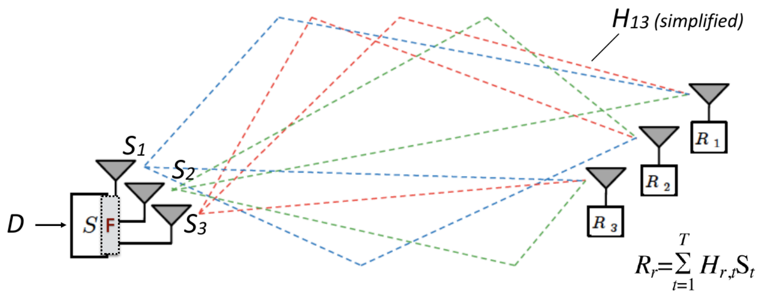

Figure 14: MIMO channel

Analysis

⊕ Stronger guarantees than SISO (beam forming focusses energy on receiver, jamming interferes with attacker)

⊖ Still no authentication

⊖ Still no guarantees on environment

⊖ Still questionable benefit over classic public-private-key schemes

⊖ Passive attack: known plaintext attack (attacker can train filter)

⊖ Active attack: abuse lack of authentication

5.3. Friendly Jamming for Confidentiality and Access Control

Friendly Jamming It consists in transmitting noise that the legitimate receiver can subtract (assuming a shared secret — possibly established with one of the earlier methods).[9]

The attacker on the other hand cannot distinguish signal and noise. However, jamming signal is much stronger and covers the spectrum of the data signal.

Example: IMD Shield for implanted medical devices.

Analysis (as claimed by the IMD Shield paper).
Let DJ be the distance between the data and jamming antenna (of the sender). **If $DJ > \lambda/2$ then an attacker with two antennas can separate the two signals** (multiple channels). Furthermore, if $DJ >> \lambda/2$ then the attacker can use directional antennas for signal separation.
Thus, the only "safe" case is $DJ < \lambda/2$ (i.e. when channels from D to A and from J to A are highly correlated).

However, this does NOT hold: a MIMO attacker can retrieve the data even when $DJ < \lambda/2$, see the following.

Attack in Line of Sight LOS Model The attacker places two antennas A, B (see Figure 15). The ideal placement is such that they receive both the jamming signal simultaneously (e.g. equidistant to the jammer) and the data signals with a phase shift of $\lambda/2$. However, even with imperfect placement some data can still be recovered (with some attenuation).

When we also consider multipath effects (additional changes in amplitude and phase offsets) the attack becomes much harder.

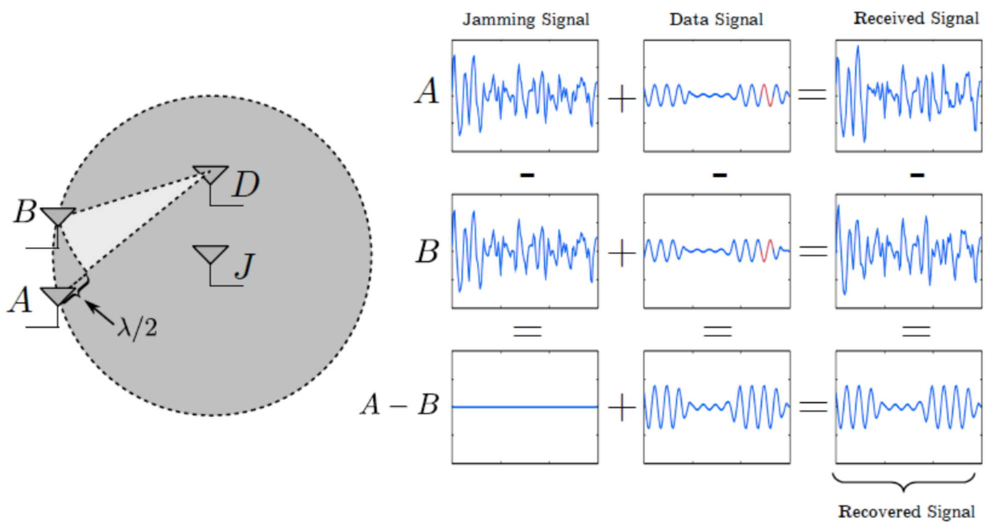

Figure 15: Friendly Jamming Attack Through Antenna Placement

[9] Note the difference to orthogonal blinding, which does not add noise to all channels but only to the null space of the receiver's channel.

Conclusion Jamming works well for access control (in the sense that J prevents malicious signals being sent to D). It does NOT work for confidentiality since a MIMO attacker can retrieve data even when $DJ < \lambda/2$.

In other words, the IMD Shield protects against malicious commands being sent to the implant, but not against an attacker listening to outgoing signals.

Signal Annihilation An attacker can deliberately introduce destructive interference to attenuate a legitimate signal. Here, we are fooling the receiver to believe that there is no signal on the channel.

Summary

- Using the physical layer for access control seems realistic.
- Using it for confidentiality however is questionable! Weak guarantees, use only as complementary measures.

6. Broadcast Authentication and Device Pairing

6.1. Broadcast Authentication: Delayed Key Disclosure

Scenario One sender, many unknown (possibly malicious) receivers. All (legitimate) receivers need to verify the authenticity of the broadcast messages.

Challenge: don't use public key cryptography (computationally expensive, especially on low-power devices), but instead only rely on symmetric cryptography.

One-way hash chain Repeated application of a hash function F, starting at an original value s_l (see Figure 16). Due to the one-way property of hash functions an attacker knowing s_i can only compute the subsequent value in the chain (s_{i-1}) but not the value s_{i+1} that generated s_i. This allows us to "use" the values s_i one-by-one and always reveal a single new value. See also Merkle hash trees.

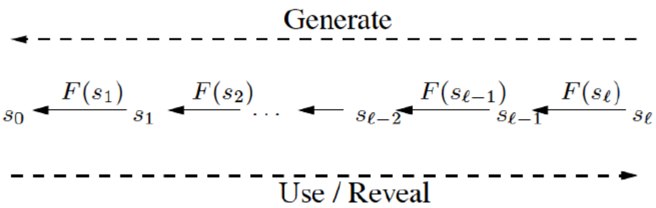

Figure 16: One-way Hash Chain

Delayed Key Disclosure (TESLA) See Figure 17.

- The sender randomly generates a secret K_l, computes the entire hash chain and distributes K_0 as a "public key" to all receivers.
- To send a message M_j in the interval i, it sends

$$P_j = \{M_j || MAC(K_i', M_j) || K_{i-d}\}$$

  where $||$ denotes concatenation and usually $d = 1$. Note that we can send multiple messages in the same time interval using the same key.
- To verify a message, the receiver needs to receive M_j and — at a later time step and in another message — the key K_i. They can then compute K_j'.[10]
  Then the receivers verify whether the MAC matches and the key K_i does correctly hash down to the pre-loaded K_0 and whether the message was received in an interval where the key was valid.
- If a key is used after the interval, the message is ignored.

Analysis: TESLA achieves asymmetry by delaying the explicit disclosure of the self-authenticating keys in clear-text. It requires time to be loosely synchronized between the sender and receivers.

6.2. Device Pairing

Scenario We want to establish a secret key between two wireless devices, in the presence of an adversary.

[10] Why this extra hashing step is needed is "an implementation detail".

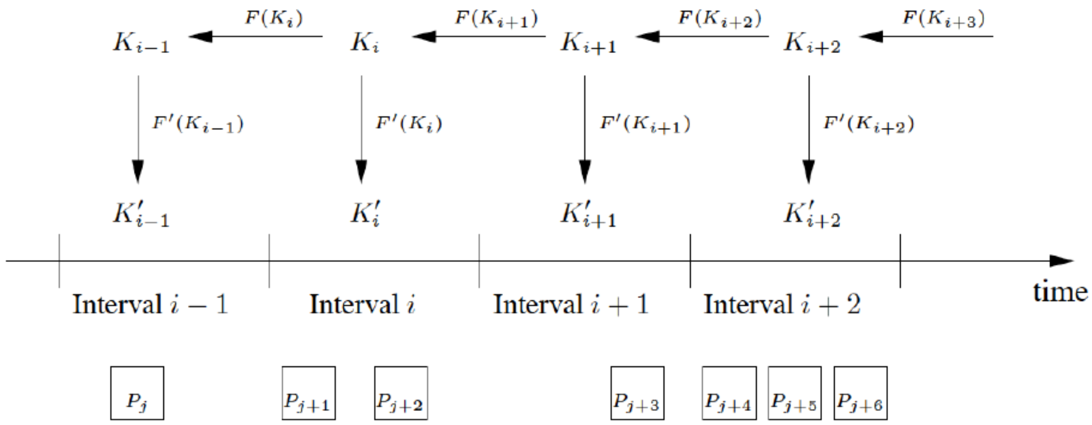

Figure 17: TESLA

Diffie-Hellman Key Exchange The classic secret key establishment scheme, established a shared key $k_{AB} = g^{ab} \mod p$. See Figure 18 for a refresher.[11] However, it is only secure against a passive adversary: an active attacker can trivially MITM the key exchange.

Goal: Authenticate the DH key exchange between two wireless devices, to ensure they have established the same key.

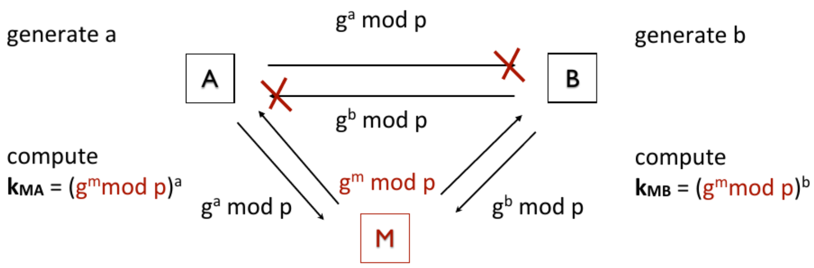

Figure 18: Diffie-Hellman Key Exchange and MITM Attack

Device Pairing Proposals A selection:

- Short string comparison: hash the key, display the string on both devices.
- Seeing is Believing: One device scans a QR code of the other's public key (one-way authentication).
- Loud and Clear: has public key, map it to a recognisable sentence, read it out loud (text-to-speech TTS).
- Integrity Regions: use distance bounding to authenticate the key if the devices are in close proximity.
- Resurrecting duckling: first to receive the key becomes the owner for life.
- Shake them up: see below.
- PIN/Passkey entry: Bluetooth.

[11] We have prime p, a generator g of Z_p^* and the Diffie-Hellman assumption stating that the discrete logarithm is hard in some groups.

Note the different security assumptions: only friends can be close, trust on first use, trusting the human, binding the success of the authentication to the happening and success of authentication (eliminating human error).

Shake them up

Idea: Divide time into N slots. In each slot, randomly either A or B transmit a message. A message is either 1 ("I am Alice") or 0 ("I am Bob"). Depending on who sent the message, this statement is either true or false. Of course, both Alice and Bob know which is the case. If the statement is true, set the next bit-to-be-exchanged to 1 otherwise to 0, thus creating a shared secret.
Synchronisation and key exchange are initiated through physically shaking the device (hence the name).

Analysis: Assumes that Eve is too far away to distinguish messages from A and B, and can thus not know whether a statement was true or false. This assumption may NOT hold as it can be attacked using signal fingerprinting to distinguish the source of a signal.
Eve can insert messages, but then A and B will set their bits to different values (since none sent the message and will assume a false statement), thus only creating a DoS situation.

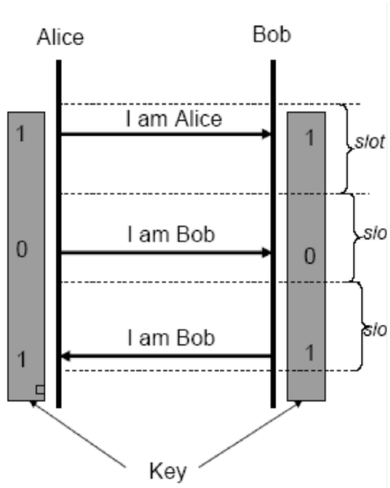

Figure 19: Shake Them Up

7. Secure Ranging

Secure Ranging, or distance measurement, is commonly used, for example, in wireless car keys, contact tracing in a pandemic, payments, automation.

Replay attacks are an issue, allowing an attacker to make devices appear physically closer (e.g. if the device naively use the observed signal strength to derive the distance).
Goals: (Provably) secure ranging, protecting against all logical and physical attacks and all attacker abilities, with a focus on preventing distance reduction, that should bind distance to identity.

Current techniques (overview)

Non-Time-of-Flight:

- Received Signal Strength Indication [RSSI] (WiFi, Bluetooth, 802.15.4, NFC, RFID) — *insecure*
- (Multi-carrier) phase measurement[12] — *insecure*
- Frequency-Modulated Continuous-Wave FMCW — *insecure*

Time-of-Flight:[13]

- Chirp Spread Spectrum (802.15.4 CSS) — *insecure*
- Ultra Wide Band UWB (802.15.4z) — *proposed*
- WiFi 802.11az, 5G — *first efforts to secure OFDM-based*

Model On a logical level, we have a **verifier V** and a **prover P**, between which we want to measure the distance. A **malicious party M** attacks this.

Additionally, we assume the worst case for the users but the best case for the attacker (bad channel/noise/multipath versus perfect channel → attacker guesses will look like noise).

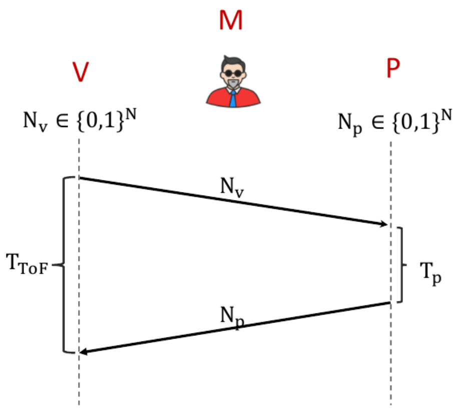

Figure 20: Model of the distance bounding scenario

[12]The distance is proportional to the phase.

[13]Calculate via $d = c \cdot (t_{tof} - t_{proc})/2$ where t_{tof} is the time between sending and receiving the signal and t_{proc} is the known processing time on the responding device.
In general, manipulating time is harder than manipulating signal properties (strength, phase).

Types of attacks (frauds)

- **Distance fraud:** A dishonest P tries to change its distance to V.
- **Mafia fraud:** Honest V, P being attacked by an external M.
- **Terrorist fraud:** Dishonest P and M collude to change P's distance.
- **Distance hijacking:** Dishonest P leverages an honest P to change its distance.

Logical Layer Distance reduction prevented using challenge-response protocols with time measurements (i.e., distance-bounding). V and P share a secret key unknown to M. V and P exchange messages that are (in part) unpredictable to M. STS cannot be predicted by M (before it is sent).

Figure 21: Scrambled Timestamp Sequence

Distance Commitment Distance commitment is the Time of Arrival measured over a public preamble and then verified using payload pulses generated using UWB pulse reordering. The timing of the preamble is binding. An attacker needs to advance payload if he advances the preamble. It's the way of knowing the distance between transmitter and receiver. As the attacker needs to advance the payload as well, he needs to guess and send before the legitimate pulse arrives, which gives him immediate feedback, and the attacker can then adjust the subsequent pulses in order to reach the needed net energy level.

Distance Bounding A prover who wants to prove its position should quickly receive N_V, compute $Ff(N_V, N_P)$ and send it. The verifier estimates prover's processing time $= t_p$. If the attacker's processing time is 0, then he can cheat by $t_p/2$. Thus, ideally, $t_p = 0s$; usually, though, $t_p = 1 - 2ns$ which means that the attacker can only shorten the distance by $15 - 30cm$. An evolution of this is

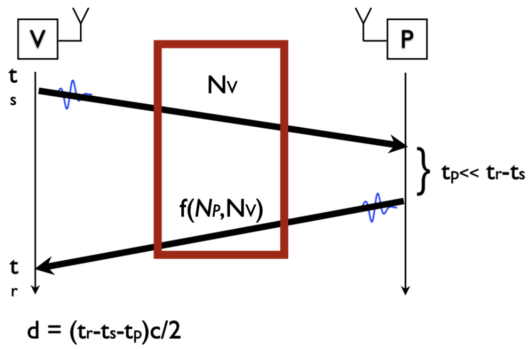

Figure 22: Distance Bounding

Challenge reflection with Channel Selection, where the prover only needs to reflect N_V and the verifier does all the work. This reduces t_p to 0.

Physical Layer: Representing bits as pulses (UWB) There are two design options to represent a bit: either with a single strong pulse or a sequence of weaker pulses. Single pulses may not be detected reliably (distance, interference), so the aggregate over several pulses is the preferred representation.

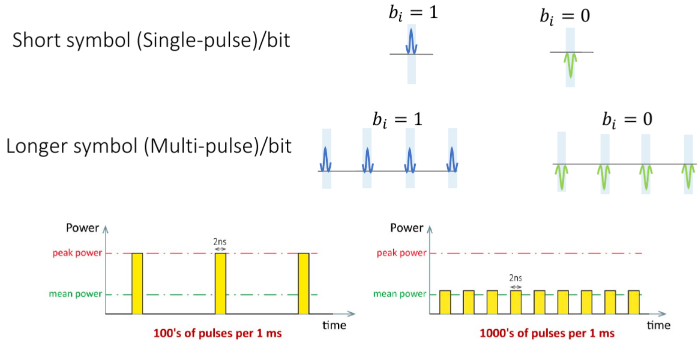

Figure 23: Bit representation: single- versus multi-pulse

7.1. Early-Detect/Late-Commit attack (ED/LC)

This attacks shortens the distance, since the receiver receives the first symbols earlier than it should have.

1. Attacker sends noise (at time T_A)
2. Attacker learns correct symbol (at time T_{ed})
3. Attacker commits to correct symbol (at time T_{lc}), by sending the remaining pulses such that the sum over all pulses matches.

Note that this attack is not possible with single pulses. A single pulse is usually $1-2$ ns long, so the attacker can cheat by at most $15-30$ cm (performance/security tradeoff).

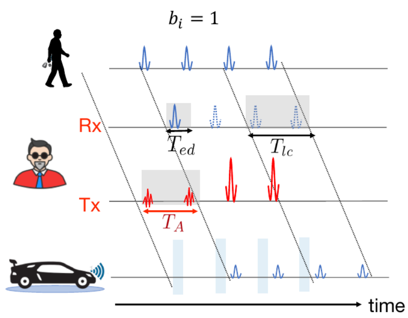

Figure 24: Early-Detect/Late-Commit attack (ED/LC)

ED/LC Solution 1: Pulse Reordering UWB-PR Interleave pulses of subsequent symbols according to some cryptographic reordering. Thus the start and end time of a symbol is unpredictable, and the attacker can only guess.
The probability of an attack's success decreases with the increase of the number of reordered bits.

ED/LC Solution 2: Variance Based Detection Statistically analyze the received versus the expected pulses. This forces the attacker to "guess better" to reduce the variance and make their error indistinguishable from the noise.

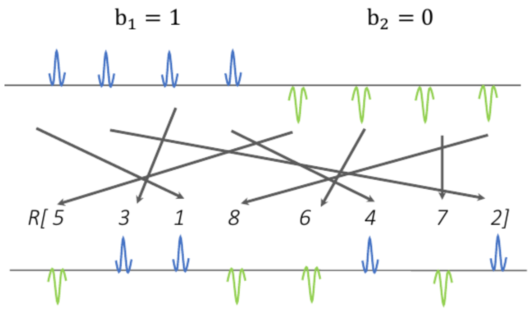

Figure 25: Pulse Reordering

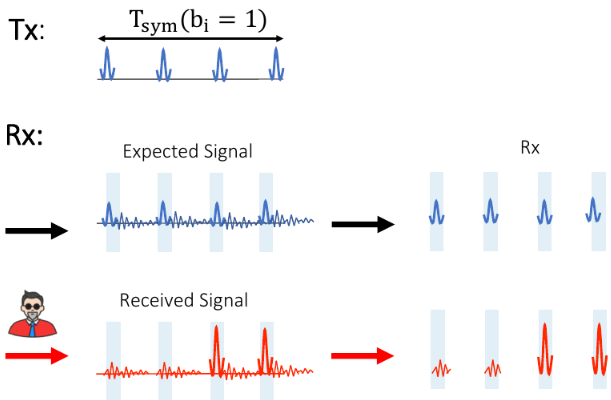

Figure 26: Variance Based Detection

ED/LC Solution 3: Scrambled Timestamp Sequence After the preamble (high correlation) which is used for ToF *estimation*, send a Scrambled Timestamp Sequence STS (encrypted with a shared secret, low autocorrelation) to use for ToF *verification*.
See IEEE 802.15.4z. Security not formally proven and unclear!

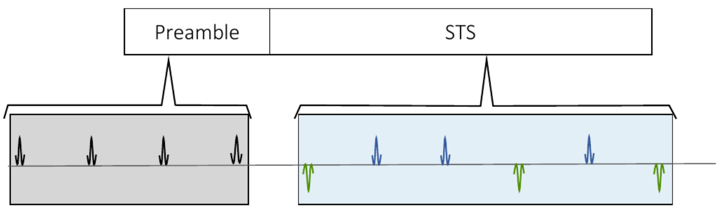

Figure 27: Scrambled Timestamp Sequence

7.2. IEEE 802.15.4z LRP versus HRP

In HRP, receivers cannot decode individual pulses. Instead, they need to find the maximum autocorrelation peak (which is not the correct ToA) and then look for the Early Path (leading edge/correct ToA)

Low Rate Pulse LRP	High Rate Pulse HRP
Can use single pulse	No single pulse (energy too low)
Multi-pulse with UWB-PR efficient	UWB-PR + variance-based seem inefficient
Open and simple security specs	No open security analysis
Low-cost, low-energy	More transparency needed

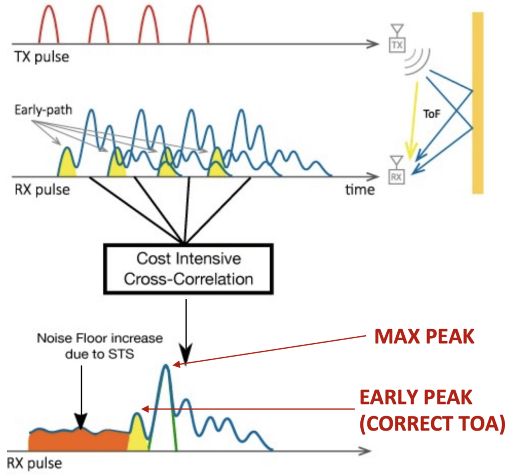

Figure 28: Early Peak

7.3. Ghost-Peak Attack

This attack shows that physical-layer attacks on HRP are a real threat. Apple/NXP/Qorvo use hidden ToA estimation algorithms making it hard to evaluate if their HRP ranging systems are truly secure. Instead, these algorithms should be public and comprehensively evaluated. Like with ciphers/signatures/MACs, their security should rely only on the unpredictability of the STS, not on the secrecy of the algorithms. It resulted in a successful reduction of the perceived distance of $4 - 12$ meters.

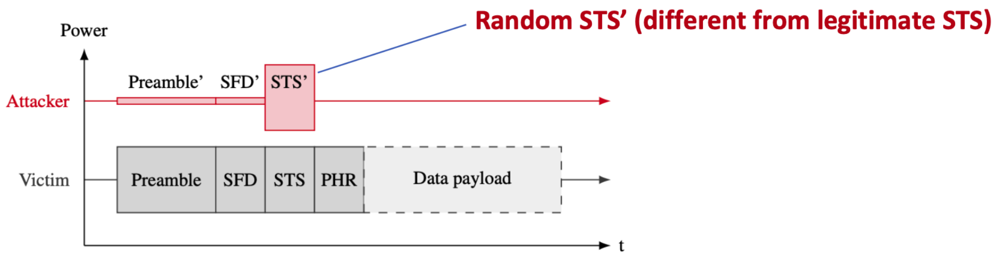

Figure 29: Ghost Peak Attack

Main idea The attack basically consists in injecting a packet in-sync (micro-seconds) with the legitimate transmission but with varying power of Preamble, SFD, STS. The attack is based on the assumptions that there is no knowledge of the keys shared between victim devices and the attacker is not able to predict the Scrambled Timing Sequence (STS). The necessary equipment is inexpensive and easily purchasable.

Message Time of Arrival Code MTAC New class of cryptographic primitives that verify the integrity of message arrival time. E.g. single-pulse, UWB-PR, Variance-based detection.

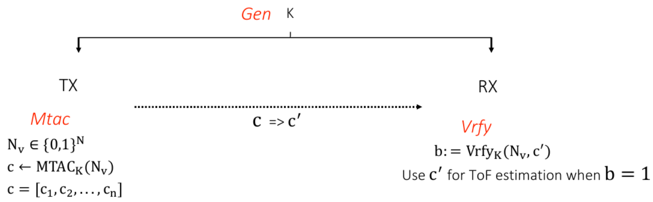

Figure 30: Message Time of Arrival Code MTAC

Verifiable Multilateration Multiple verifiers with known locations want to determine the position of a prover. C.f. GPS trilateration. E.g. to position autonomous cars using cell towers.

Future Work

- Secure Positioning (e.g. verifiable multilateration)
- WiFi 802.11 and 5G ranging (some initial work)
- Efficient implementation + deployment

8. WiFi Security

8.1. WiFi Basics

Terminology

- *Station (STA)/client*: terminal with access to the wireless media
- *Access Point (AP)*: Station integrated both with the wireless media and the distribution system
- *Service set/extended service set ESS*: Group of nodes logically grouped together and identified by their SSID
- *Basic service set (BSS)*: Subgroups of a service set using the same radio frequency/the same AP (or in general: the same physical layer medium access means).
- *Service set identifier (SSID)*: 32 byte name (usually human-readable) to identify a network
- *Channel*: 20 MHz wide frequency range to use for WiFi, typically around 2.4 GHz and 5 GHz (see Figure 2).
- *Medium Access Control (MAC)*: Goal: deliver data reliably and securely while sharing the open medium

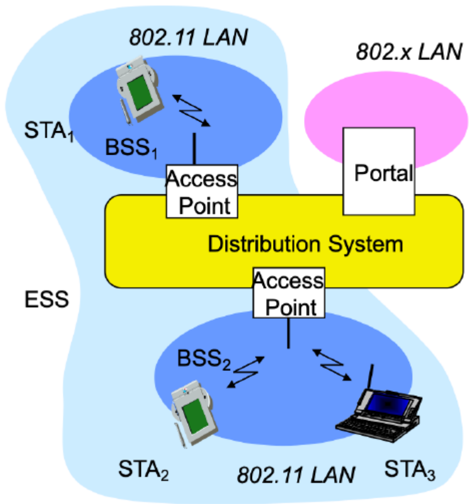

Figure 31: WiFi System

Carrier-Sense Multiple Access with Collision Avoidance CSMA/CA Medium access control mechanism. Part of the *Distributed Coordination Function DCF*.

1. *Carrier sense* — monitor medium to check if it is idle
2. *Collision avoidance* — if another node sends, wait for a randomised *backoff period*, then listen again
3. *Transmit* — send entire frame, wait for ACK, if no ACK then backoff and wait

Standard	802.11a	802.11b	802.11g	802.11n	802.11ac	802.11ad
Year introduced	1999	1999	2003	2000	2012	2014
Maximum data transfer speed	54 Mbps	11 Mbps	54 Mbps	65 to 600 Mbps	78 Mbps to 3.2 Gbps	6.76 Gbps
Frequency band	5 GHz	2.4 GHz	2.4 GHz	2.4 or 5 GHz	5 GHz	60 GHz
Channel bandwidth	20 MHz	20 MHz	20 MHz	20, 40 MHz	40, 80, 160 MHz	2160 MHz
Highest order modulation	64 QAM	11 CCK	64 QAM	64 QAM	256 QAM	64 QAM
Spectrum usage	OFDM	DSSS	DSSS, OFDM	OFDM	SC-OFDM	SC, OFDM
Antenna configuration	1×1 SISO	1×1 SISO	1×1 SISO	Up to 4×4 MIMO	Up to 8×8 MIMO, MU-MIMO	1×1 SISO

Figure 32: WiFi Standard Versions

Hidden terminal problem Occurs if a node B can communicate with two nodes A, C that cannot communicate with each other. That is, if they both tried to send data to B they would individually sense the medium to be idle, but then they would clash at B anyway. See Figure 33.

The solution is to send a *Request-to-Send RTS* message before transmitting a frame and wait for all nodes to reply with a *Clear-to-Send CTS*. Now when A sends a RTS even though C does not receive it, C still receives the CTS from B and will thus back off.
This feature is optional in IEEE 802.11!

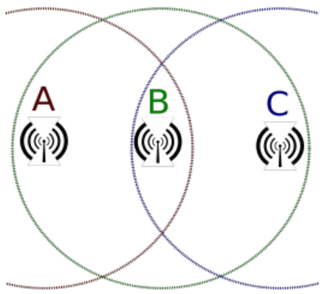

Figure 33: Hidden Terminal Problem

Frame types Data frames (user traffic), control frames (ACK, RTS/CTS), management frames (beacon[14], association request, deauthentication).

[14]Contains basic information about the network, e.g. SSID, FH parameters, DSSS parameters, other supported options.

8.2. Basic Manipulations

Capturing all frames Easy — just install any packet capture software (such as Wireshark) and set the wireless interface to "promiscuous mode".

Communication (un)fairness DCF is fair — assuming all stations adhere to it! However, by modifying backoff parameters in the wireless card driver, one can behave selfishly to get a higher throughput.

Jamming
Simple:
(a) Trigger carrier sense of transmitter (prevent sending, less power).
(b) Mangle the frame at the receiver (prevent receiving, more power).

Selective:
Listen, decode prefix of an incoming frame and based on recipient address decide whether to jam the remaining data portion of the frame. One needs to be very fast!

Man-in-the-middle Clone real AP onto different channel → get clients to sign up to yours (e.g. jam real AP, send a Channel Switch Announcement CSA) → inspect/manipulate/forward frames to real AP → profit.

8.3. WiFi Security Standards

Wired Equivalent Privacy WEP

First WiFi security standard (1997), with the goal of making wireless as secure as wired networks. Today fully broken.

Operation:
Checksum: plaintext $P = (m, CRC(m))$
Encryption: ciphertext $C = P \oplus RC4(IV, k)$ (stream cipher RC4 with shared key k, initialisation vector IV)
Transmit (IV, C).

Problems:

- *Confidentiality*: Keystream reuse allows recovery of plaintexts:

$$C_1 \oplus C_2 = (P_1 \oplus RC4(IV, k)) \oplus (P_2 \oplus RC4(IV, k)) = P_1 \oplus P_2$$

  Since the IV has 24 bits, at a data rate of 5 Mbps it will repeat after ≈ half a day. Knowing one of the plaintexts is reasonable (common structures such as IP headers, packet injection from the internet, other redundancies).

- *Integrity*: CRC-32 is not a cryptographic MAC. This allows controlled message modification, i.e. given a $(ciphertext, checksum)$ pair we can create another $(ciphertext', checksum')$ pair such that the checksum is valid for the decrypted $plaintext'$.

- *Access control*: works as follows:

  1. AP sends a plaintext challenge
  2. Station replies with WEP encryption of challenge (proving possession of the key)

3. AP completes association request

Through simple traffic capture an adversary can learn a valid plaintext/ciphertext pair. From that they can derive the keystream to later compute a valid response to a new challenge (since the station can chose the IV).

Since 2007, tools like aircrack-ng allow recovering WEP keys in a matter of minutes.
Wrong idea during design: “in-flight” modification of packets may be hard, but don’t forget the possibility of offline attacks!

WiFi Protected Access WPA / Temporal Key Integrity Protocol TKIP

Designed as a transitional mechanism in 2003 and to be compatible to WEP devices (though in 2019 still ≈ 50% of networks accept it. . .).

Idea:
Prevent keystream reuse and use a cryptographic MAC, while staying compatible to WEP. “Achieved” by augmenting encryption with per-packet key mixing, RC4 keystream filtering, a new integrity mechanism (MICHAEL) and counters for replay protection (TSC). This allows to keep the WEP mechanism in hardware but change the inputs given to it.

ChopChop attack: (for WEP!)
Capture an encrypted WEP frame. Remove the last data byte, guess it, re-compute checksum for the shorter packet, use AP as an oracle (must discard frames whose checksums fail). Rinse and repeat to learn one data block after the other.
For a detailed explanation see this aircrack doc entry.
Since this is a stream cipher, knowing one plaintext/ciphertext pair also reveals a keystream.

A similar attack also works for WPA/TKIP since MIC(HAEL) is reversible and the AP reports failures.

Biased keystream:
There exist statistical biases in a RC4 keystream. With enough packets this can be used to recover the plaintext for one MIC key. Computationally expensive, but shows that crypto can be attacked, too.

Summary:
Difficult start (had to reuse WEP hardware). Raised the bar (despite ugly fix), but attacked in 2009, 2014/2015. Tolerable transition mechanism.

WPA2

Introduced in 2004. Better encryption (AES-128) and integrity (CBC-MAC, using *authenticate-then-encrypt*). Better authentication with a 4-way handshake.

Handshake:
Goal: Starting from a *Pairwise Master Key PMK* (that is derived from a shared passphrase) provide mutual authentication and establish a session key (*Pairwise Transient Key PTK*).
Steps: exchange nonces (msg1,2), derive PTK using PMK + nonces + MAC addresses, authenticate exchange (msg3,4). Note that in msg2+3 the *Message Integrity Code MIC* is also transmitted (missing in Figure 34).
Proven “secure” in 2005 (mutual authentication and password is not leaked).

Dictionary attack:
An adversary can capture the ANonce, SNonce and MIC. *If* the user chose a weak passphrase, then

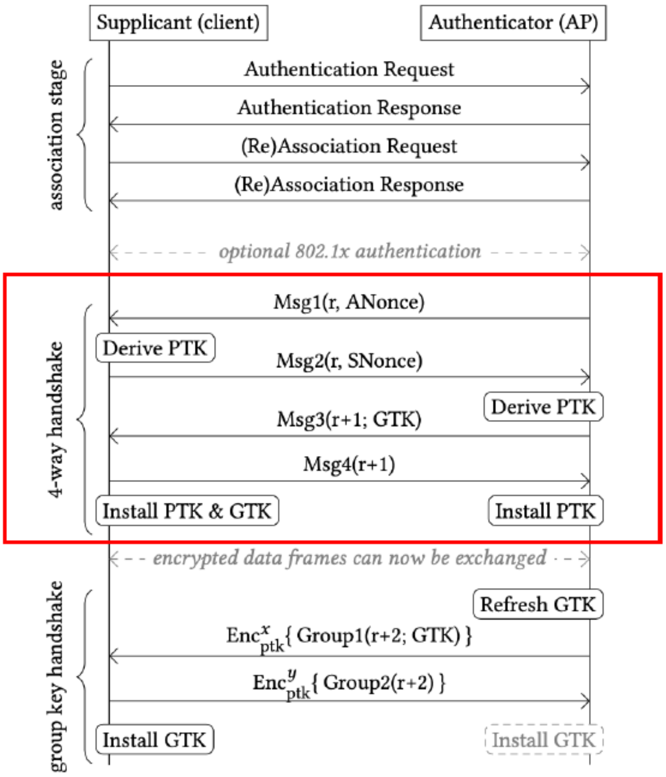

Figure 34: WPA2 Handshake

by brute force an attacker can try different passphrases, compute different PTKs and thus different MICs and compare the result to the captured MIC.

Key Reinstallation Attack (KRACK) (2017):
Goal: force a keystream reuse.
Observations: AP may retransmit msg3 if no ACK. Each time, client reinstalls the *same* PTK, and in the process it resets some counters. This resulting in the same keystream being reused.
Approach: Replay msg3.

Summary:
Solid cryptography/encryption, weakness in handshake. Proven properties hold, but model does not capture *when* a key is installed. Can be patched (and re-attacked).

WPA3

Introduced in 2018. Updated cryptography (AES-128/256 encryption, SHA-384 HMAC integrity), new handshake (based on the Dragonfly Key Exchange (RFC 7664)).

Handshake improvements:
Turns a low-entropy password into a high-entropy key, thus allowing for shorter passwords. Has forward secrecy.

Dragonblood (2020):
Transition mode allows downgrade to WPA2.
Timing attack: execution time of turning the password into a group element at the start of the handshake depends on the password (i.e. a side-channel attack, though only for bad curves).

FragAttacks Several implementation flaws can be abused to easily inject frames into a protected Wi-Fi network. In particular, an adversary can often inject an unencrypted Wi-Fi frame by carefully constructing this frame. This can for instance be abused to intercept a client's traffic by tricking the client into using a malicious DNS server as shown in the demo (the intercepted traffic may have another layer of protection though). Against routers this can also be abused to bypass the NAT/firewall, allowing the adversary to subsequently attack devices in the local Wi-Fi network (e.g. attacking an outdated Windows 7 machine as shown in the demo).

How can the adversary construct unencrypted Wi-Fi frames so they are accepted by a vulnerable device? First, certain Wi-Fi devices accept any unencrypted frame even when connected to a protected Wi-Fi network.

Additionally, certain devices accept plaintext aggregated frames that look like handshake messages. An adversary can exploit this by sending an aggregated frame whose starts resembles a handshake message and whose second subframe contains the packet that the adversary wants to inject. A vulnerable device will first interpret this frame as a handshake message, but will subsequently process it as an aggregated frame. In a sense, one part of the code will think the frame is a handshake message and will accept it even though it's not encrypted. Another part of the code will instead see it as an aggregated frame and will process the packet that the adversary wants to inject.

Design flaw: aggregation attack

The first design flaw is in the frame aggregation feature of Wi-Fi. This feature increases the speed and throughput of a network by combining small frames into a larger aggregated frame. To implement this feature, the header of each frame contains a flag that indicates whether the (encrypted) transported data contains a single or aggregated frame. This is illustrated in the following figure:

Unfortunately, this "is aggregated" flag is not authenticated and can be modified by an adversary, meaning a victim can be tricked into processing the encrypted transported data in an unintended manner. An adversary can abuse this to inject arbitrary network packets by tricking the victim into connecting to their server and then setting the "is aggregated" flag of carefully selected packets. Practically all tested devices were vulnerable to this attack. The ability to inject packets can in turn be abused to intercept a victim's traffic by making it use a malicious DNS server.[15]

[15] fragattacks.com

9. Cellular Security

9.1. 1G — Analog

Overview Introduced in the early 1980s to connect to the telephone network (*Public Switched Telephone Network PSTN*). Medium access control: split bandwidth with FDMA, with one call using the same frequency in both directions. Suppprts handover between different base stations.

No security Identification via serial and phone numbers. Control messages as analogue tones.

Problems: eavesdropping (privacy), mobile cloning (billing fraud).

9.2. 2G — GSM

Overview Introduced in the early 1990s. Digital voice and control messages, enabling features like: compression, error correction, less power, SMS, security mechanisms. We focus on the *Global System for Mobile Communications GSM*.

Architecture See Figure 35.

Medium access control: FDMA with distinct uplink/downlink frequency channels. TDMA[16] to support 8 speech channels on the same frequency.

Different channels for traffic and control frames (e.g. paging channel, random access channel, access grant channel).

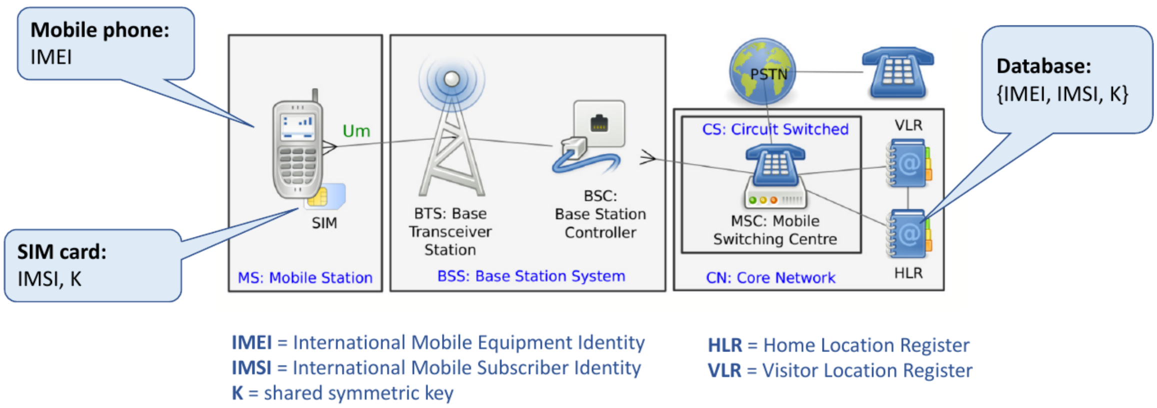

Figure 35: Architecture of 2G

Security model The main goal was to prevent the misuse of a subscriber identity by a third party. Everything based on symmetric shared keys K_i. The key is stored in the *Home Location Register HLR* of the provider and on the *subscriber identification module SIM* card (and never leaves it).

Algorithms: A3 for authentication, A5 for encryption, A8 for key derivation. Initially secret, but A5 leaked in the mid 90s, and got reverse engineered in 1999.

[16] Time-division multiple access

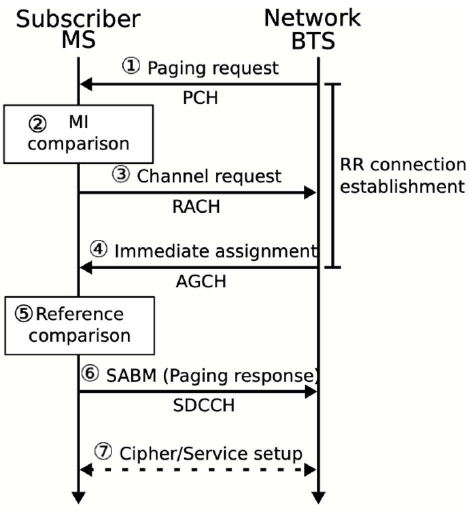

Figure 36: Setup of incoming call (BTS "pages" the MS) over the different channels

GSM authentication See Figure 37. From the shared key K_i and a random challenge a session key K_c is derived. Together with the frame nonce/counter F_n it is used to encrypt the plaintext frame m_i.

Note that there is no mutual authentication (only the phone is authenticated) and messages can be replayed! Also note that since A3 and A8 are executed on the SIM card, the operator can choose these!

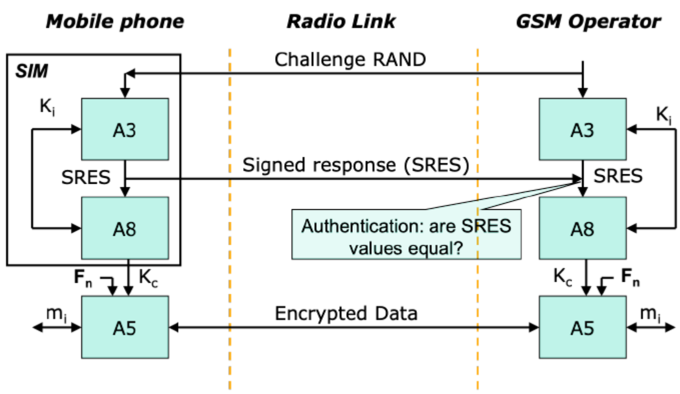

Figure 37: 2G (Authentication) Flow

GSM encryption Goal: fast in hardware. Two variants A5/1 (strong) and A5/2 (weak, not discussed here).

A5/1: stream cipher with a *Linear Shift Feedback Register LSFR* and 64 bit security. Registers are initialized with the key K_c and the frame counter F_n to create the keystream, which is then XORed with the plaintext. See here for a visualization of the LSFR.

Attack approach:

Known plaintext/ciphertext pair $\overset{XOR}{\longrightarrow}$ keystream $\longrightarrow$ secret internal state $\overset{solve\ LSE\ with\ 64\ eqns}{\longrightarrow}$ key

A5/1 attacks Attacks of A5/1 evolved over time (2000–2010). The first were not very practical (requiring many known plaintexts or special-purpose hardware). Types included: Time-Memory Tradeoff

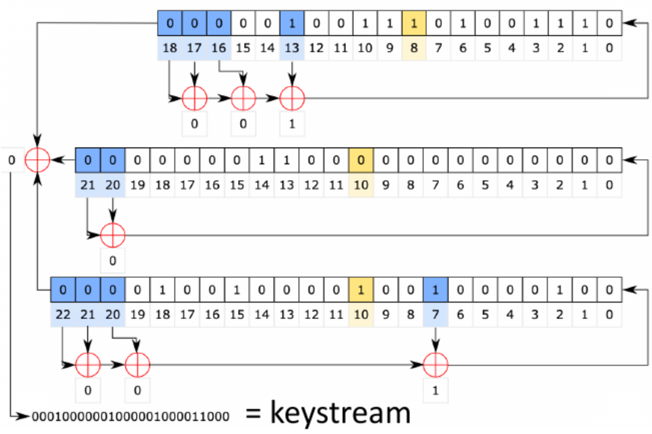

Figure 38: 2G A5 Linear Shift Feedback Registers

Attack (Biryukov 2000, Nohl 2010), Correlation Attack (Brakan & Birham 2005), Guess and Determine Attack (Gendrullis et al. 2008), Fast Near Collisions (Zhang 2019).
Note that a few known plaintexts are reasonable (known control frames).

Karsten Nohl (2010): 2TB precomputed table (mapping from keys/state to keystreams, 1 month computation with 4 GPUs), 64 bit plaintext, 5 sec attack time (lookup).[17]
Challenge: reducing table size (2^{64} unreasonable).

Rainbow tables (Oechslin 2003)
Solves issue of collisions in chains (leading to reduced keyspace coverage as chains merge). Uses different variant f_i ("color") for each chain link. Different lookup details, but same idea.

A5/1 attacks summary Previous ideas generally applicable to stream ciphers, not just to A5/1. Main enabler is the short key size (64 bit). Nevertheless it lasted quite well considering when it was designed and under which hardware constraints, and given that there is still ongoing research.

A8 attacks Setup: K_c known, want to recover K_i.
1998: COMP128 hash inverted in hours (effectively only 54 bit)
2002: Faster recovery using side-channels.
Mitigation: Operators replace A8 (OTA update, new SIM).

GSM — no integrity protection No integrity protection defined, due to too much overhead (voice frame has 144 bits). Also special use case: dropping frames/retransmission is undesired, and small voice frame modifications are acceptable.

GSM — no mutual authentication Recall that the phone does not authenticate the base station. Probable reasoning (1980s): expensive equipment, call encrypted anyway. But: commercial fake BS (2000s), USRP (2010), etc enable user identification + tracking and MITM.

[17] Nohl omitted some attack details, later provided by Lu (2015).

9.3. 3G — UMTS

Overview *Universal Mobile Telecommunication System UMTS* introduced in the early 2000s.

Radio link uses wideband code-division multiple access W-CDMA, separate per-user spreading codes, distinct uplink and downlink frequency bands.

Protocol New **authentication and key agreement (AKA)** protocol. Provides mutual authentication, mutual replay protection, integrity protection. Also used in 4G+5G (more or less).

Similar design principles like GSM: operator and SIM trusted, phone and visited networks untrusted, minimize communication with home network.

Remarks: (see Figure 40)

1. Both the SIM card and the operator maintain the sequence number SQN (against replays)
2. The IMSI[18] is send before the authentication (enabling tracking, see later)
3. Function f_1, f_2, f_3, f_4, f_5 are operator-specific
4. Loose synchronisation required
5. Integrity key IK for integrity protection

Further details on the authentication, encryption and integrity protection[19] functions can be found in the slides but are omitted in this summary.

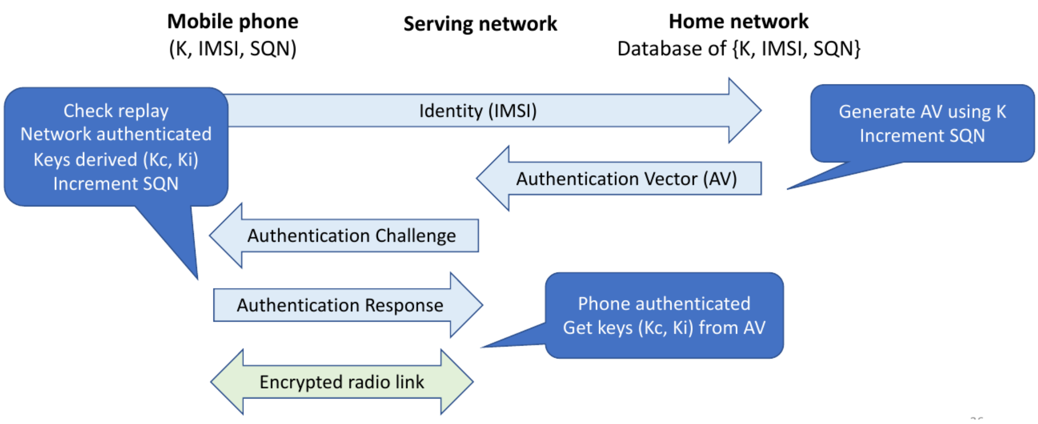

Figure 39: AKA High Level Flow

Cryptography Summary Authentication and Key Agreement protocol formally verified with respect to authentication and confidentiality (2001). Two known but impractical attacks on encryption (interesting for research though). TLDR: Good for now.

[18]International Mobile Subscriber Identity

[19]Mandatory for signalling + control messages, optional for data. Based on a 8-round Feistel network to be fast in hardware.

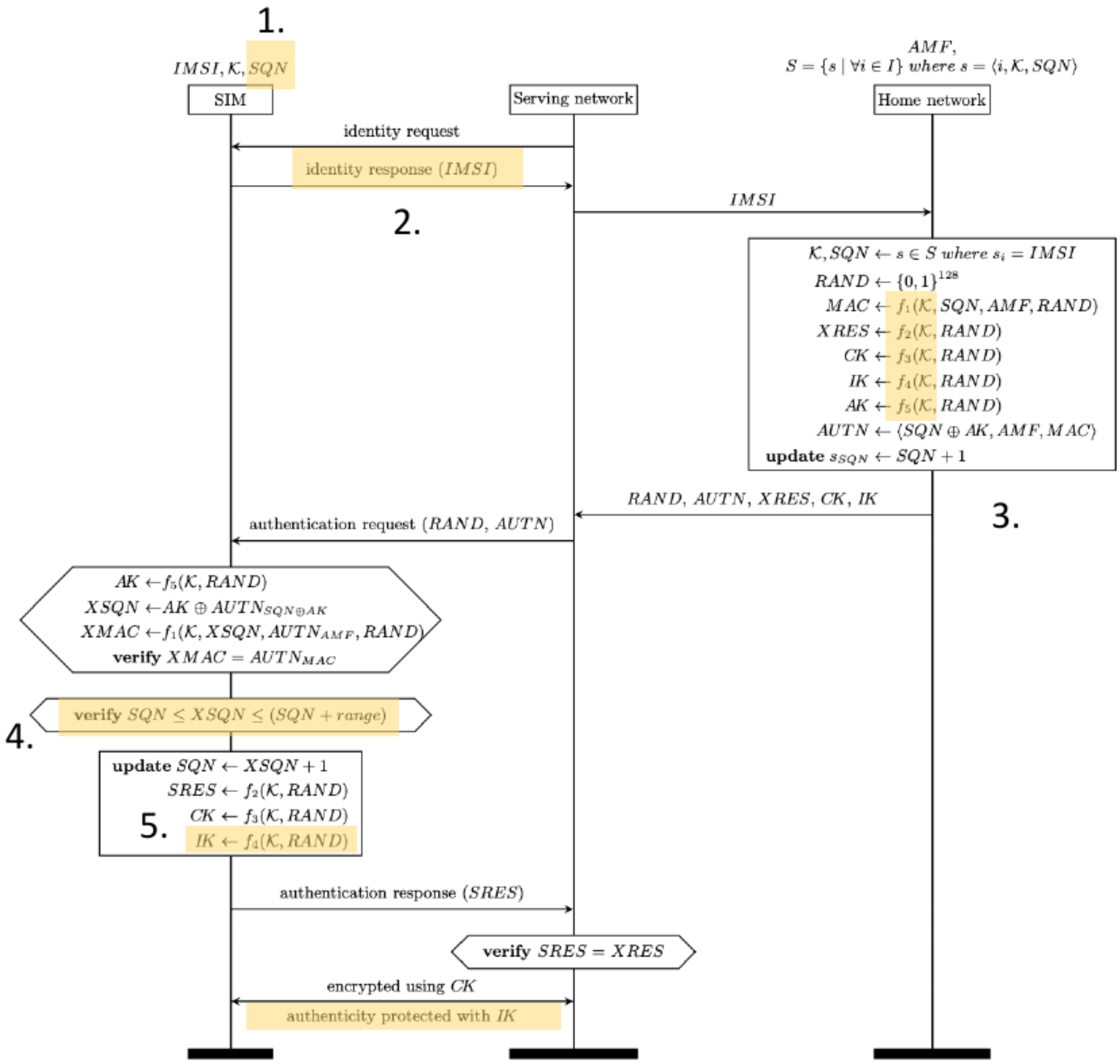

Figure 40: AKA Detailed Flow

Denial of Service Commercial jammers available for a few hundred dollars (though use is illegal!). Approaches:

- Insert noise on physical layer.
- Jam/block paging messages (control layer). Difficult, requires synchronisation with the victim.
- Answer paging messages faster than the victim, causing the victim's reply to be ignored. Possible because (a) paging occurs before authentication and (b) base stations cover large areas.

MITM/Fake BS Though AKA authentication is mutual, a MITM perform a downgrade attack to force the phone to use GSM (due to co-existence). Also, in 3G the MiTM can learn the IMSI at the start of AKA.

Practical considerations: Which frequency to use — allocated or unallocated? What cell id to use — a new, unknown one? Jam legitimate BS to get victims to connect to yours?
$\implies$ Though setting up a fake BS is easy, detecting it is easy as well.

Femtocells Operator provides a "mini-BS" to customers to improve local (indoor) coverage. The femtocell box relays from the radio link to the operator network.

Vulnerable because they are easier to access then normal base stations (high up on a tower). Gaining access gives a perfect MITM position, having the keys for the gateway to the operator as well as for the radio link.

User tracking Identity (IMSI) is sent before authentication. Even though a temporary identity (TMSI) is issued, the IMSI is reused on occasions. Thus, user tracking is possible to some extent, but not addressed by the spec (tradeoff possibility of abuse versus increased complexity).

Approaches for identity protection:

- *Pseudonyms*: send pseudonym when starting AKA, with the home network always returning a new pseudonym (encrypted[20], so that the serving network cannot read it).
  Challenge: requires synchronisation, and thus a recovery process. However, it is hard to design a recovery process that cannot be abused to learn the IMSI.
- *Public key encryption*: store home network public key on SIM, encrypt IMSI. Defined as optional in 5G. Pro: no state that needs to be synchronised. Con: asymmetric cryptography is expensive.

9.4. Signalling System 7 — SS7

Signalling network used in GSM + 3G to route calls, coordinate roaming, deliver SMS, etc. Defined in the 80s/90s.

Initially only a few participating, mutually trusted operators. But: Soon grew to 1000+ operators and third-party service providers, and SS7 access could be purchased at a low price.
$\implies$ Trust assumption violated.

Open source software and specs online, anybody with network access can send SS7 commands with a Linux computer.
$\implies$ Assumption on expensive equipment violated.

Attacks in 2014 by Engel (see the 31C3 talk here), discussed below.

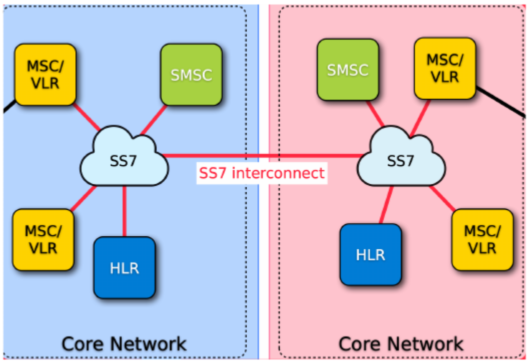

Figure 41: SS7 Architecture

[20]Note that this encryption can be done symmetrically with the shared key, since the home network could use the pseudonym to look it up.

Location tracking Phone locations are stored in the *Gateway Mobile Location Center GMLC*, access to which requires authentication (e.g. law enforcement). However, by requesting the routing info from the HLR and with that the cell id from the *Mobile Switching Center MSC* where the user is currently logged in, one can work around this to still get a rough location.

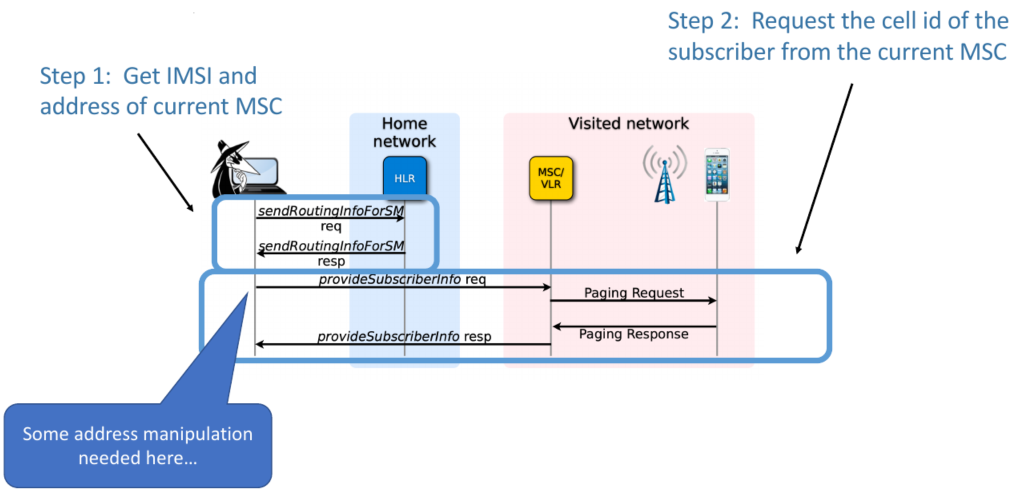

Figure 42: SS7 Location Tracking

Intercepting Calls See Figure 43.

1. Attacker overrides the **GSM Service Control Function** *(gsmSCF)* in the MSC with their own.
2. When the target makes a call, the MSC now contacts the attacker.
3. The attacker learns the phone number and rewrites it towards their recording proxy.
4. MSC sets up call to the proxy.
5. Proxy bridges call to the intended receiver.

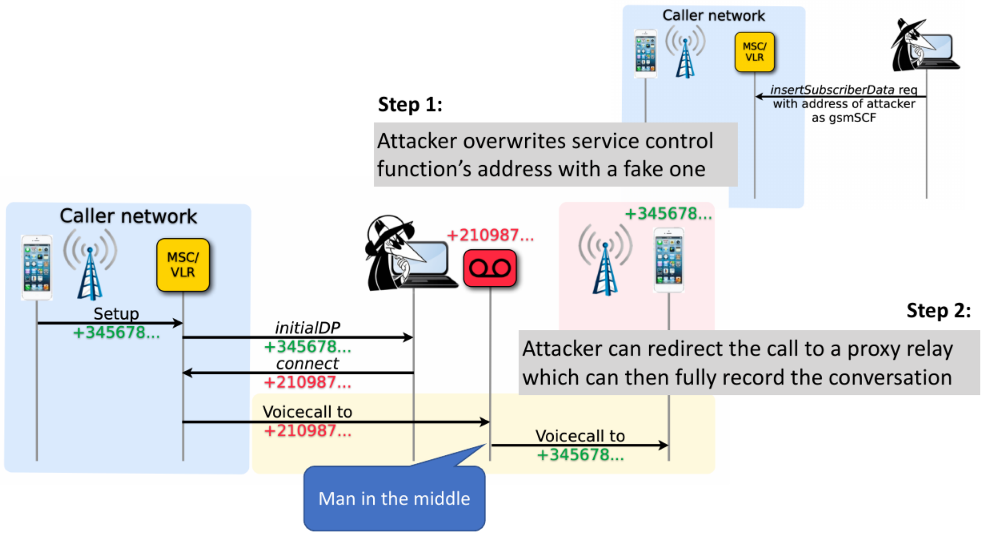

Figure 43: SS7 Intercepting Calls

SS7 Summary Legacy system with outdated trust model. Bad network management (open interfaces, no authentication or access control to control messages). Attacks are independent of cryptography and the radio link (i.e. work from far away).
Some issues were fixed, and LTE has a new signalling system (Diameter).

9.5. 4G — LTE

Overview *Long-Term Evolution LTE* introduced in 2008.
Updated architecture: fully packet switched, new core network (*Evolved Packet Core EPC*, fully packet-switched), new radio network (*Evolved UMTS Terrestrial Radio Access Network E-UTRAN*), but interoperable with legacy systems.
Updated physical layer: *Orthogonal Frequency Division Multiplexing OFDM* (downlink with orthogonal sub-carriers, single-carrier uplink), multiple antennas (MIMO).

Architecture and Terminology See Figure 44.

- *User Equipment UE* (MS): the mobile handset
- *Evolved Node B eNB* (BS): the base station
- *Mobility Management Entity MME*: handles signalling via the *Non-access stratum NAS*, UE authorisation, S-GW selection
- *Home Subscriber Server HSS* (HLR): subscriber database, user authentication
- *Serving Gateway S-GW*: routes user data packets
- *Packet Gateway P-GW*: connects to external network, routing, filtering

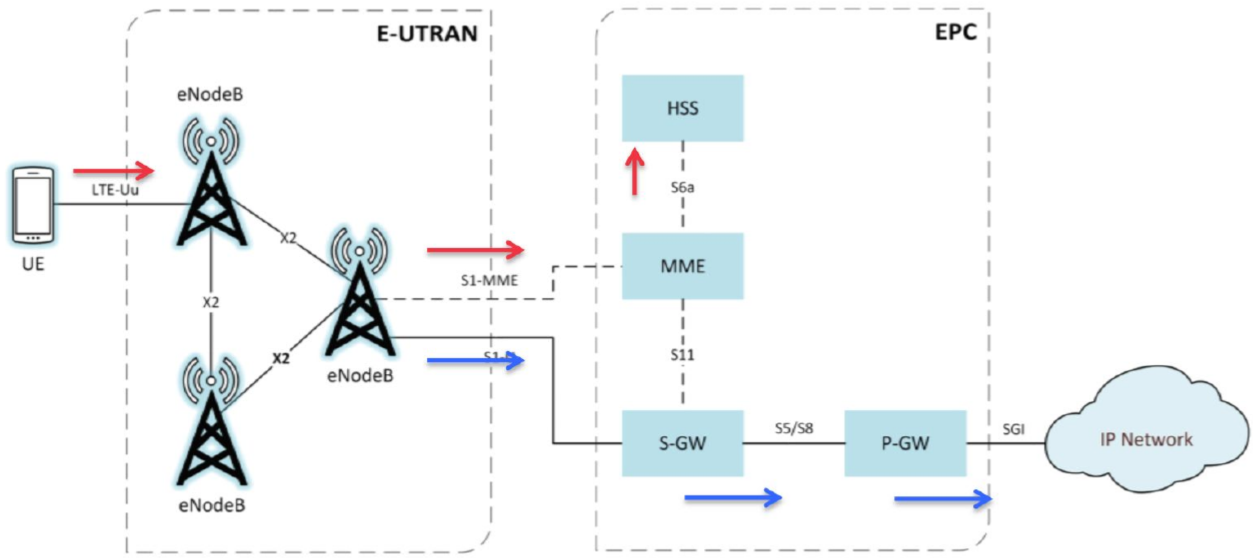

Figure 44: LTE Architecture

Network Protocol Stack See Figure 45. From top to bottom:

- *Non-access stratum NAS*: mobility management, tracking area update, etc
- *Radio Resource Control RRC*: AKA, paging messages, system information broadcast, etc.
- *Packet Data Convergence Protocol PDCP*: compression, optionally encryption+integrity
- *Radio Link Control RLC*: error correction, segmentation, frame ordering
- *MAC layer*: manages access to radio link

Note that everything below the PDCP layer is unencrypted. Thus most sniffing+spoofing attacks focus on the layers below.

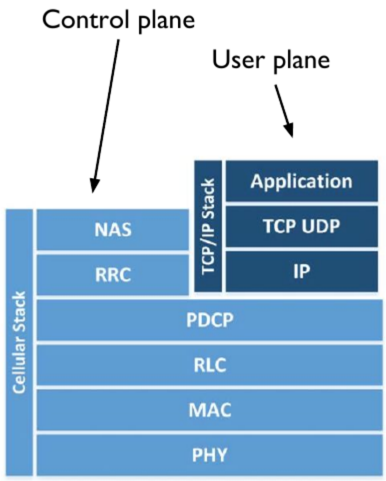

Figure 45: LTE Network Protocol Stack

Uplink congestion control

Security Overview Authentication: similar to AKA.

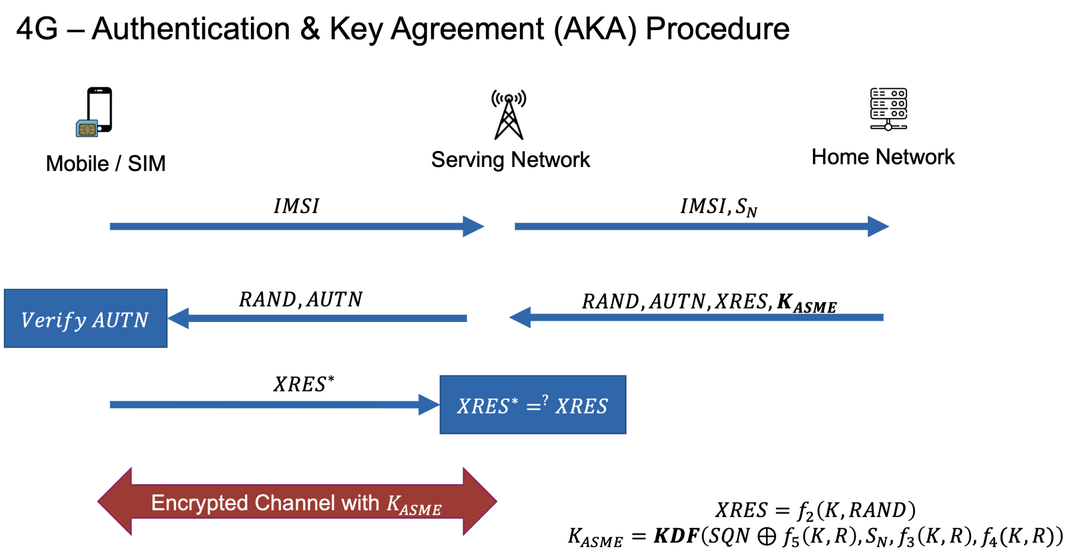

Figure 46: LTE Architecture

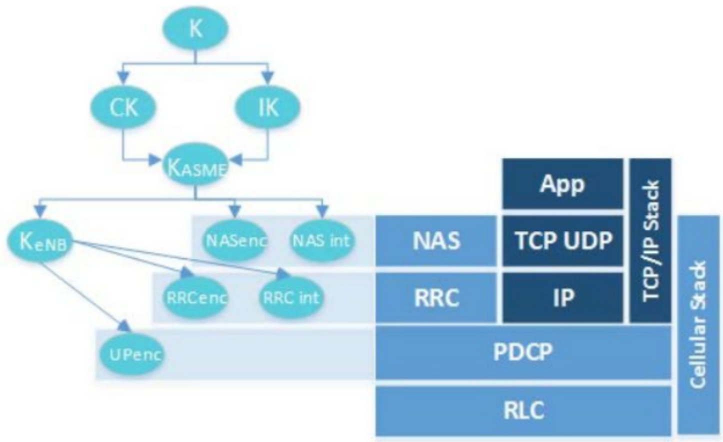

Figure 47: LTE Key Hierarchy

Encryption/integrity: 3 variants EEA1, EEA2, EEA3 and EIA1, EIA2, EIA3.

Other: extended key hierarchy, option for longer keys (256 bit), handover between eNBs (X2), backhaul (S1) protection.

Authentication is required in both control and user plane.

Encryption is optional for both the user and control plane, but often used.

Integrity protection is mandatory for the control plane, but not for user data

Key hierarchy to limit attack possibilities and impact. Master key K (128 bits, stored on HSS+SIM), confidentiality key CK, integrity key IK, etc.

Backhaul + EPC protection For backhaul, the LTE spec recommends physical protection. For EPC, the spec is vague ("division of security domains"). In practice both are secured using standard IP security practices (VPN, PKI).

Handover + Key Separation Reduced attack surface and key scope by limiting key lifetime of K_{eNB}. E.g. different keys for different eNBs/cells.

Location tracking

Background: The service area is divided intro *tracking areas TAs* containing multiple cells (each controlled by an eNodeB that broadcasts information such as the TA code, mobile network code, cell ID). UE sends IMSI with the Attach request, in which the operator assigns temporary identifiers that are used subsequently (TMSI, GUTI[21]).

Adversary: Goal: learn user locations. Capabilities: transmit/receiver radio signals, possible with commercial USRPs. Advantage: GUTI re-allocation depends on operator, possibly not changed for multiple day.

Attack:

1. Set up fake BS.
2. Monitor user presence in TA.

[21] Global unique temporary identifier

3. Learn precise location: actively send unprotected *RRC Connection Reconfig* messages, to which the UE responds with a *Measurement Report* containing the signal strengths of neighbouring cells and its GPS location.

Analysis: Not all signalling/control messages are integrity protected/authenticated. Spec allows this explicitly for troubleshooting (availability versus privacy).

MITM

Background: MAC layer assigns unique *Radio Network Temporary Identifiers RNTIs* to distinguish UEs. eNodeB uses *Downlink Control Information DCI* to notify UEs when radio resources are available. Also recall that EEA2 uses AES-CTR for encryption (XORs keystream with plaintext).

Attack:

1. Identify UE from encrypted traffic: observe connection establishment, learn TMSI+RNTI, use paging to map TMSI to phone number.
2. Modify/redirect encrypted traffic: often uplink is encrypted but not integrity protected. Xor ciphertext with "manipulation mask" (try-and-error).

Analysis: Identifiers on lower layers, encryption on higher layers. Integrity protection optional.

Jamming/DoS Brute-force jamming always possible, but requires a lot of power. Instead, targeting specific control channels can be effective, too (see next point).

Keystream Reuse Attack / ReVoLTE

Idea: IV for EEA comprised of a counter, radio bearer ID and radio direction.
Unfortunately, many operators re-use bearer IDs and reset counter for subsequent calls (exactly what we need!). Adversary can initiate a second call just after the target call and record both calls, both unencrypted and ciphered.
Fix: don't repeat id and counter.

Signal Overshadowing (SigOver)

Idea: SigOver is a signal injection attack that exploits the fundamental weakness of physical layer in Long-Term Evolution (LTE). Since LTE communication is based on an open medium, a legitimate signal can potentially be counterfeited by a malicious signal. In addition, although most LTE signaling messages are protected from modification using cryptographic primitives, broadcast messages in LTE have never been integrity protected.[22]

This attack has several advantages and differences when compared with existing attacks using a fake base station. For example, with a 3 dB power difference from a legitimate signal, the SigOver demonstrated a 98% success rate when compared with the 80% success rate of attacks achieved using a fake base station, even with a 35 dB power difference. Given that the SigOver is a novel primitive attack, it yields five new attack scenarios and implications.

Analysis: Low jamming-to-signal ratio (J/S), thus stealthy (not as obvious as a fake BS). Only downlink affected, thus undetected by the base station. Challenges: time+frequency synchronisation with the legitimate signal, distance/delay estimation to the UE, phone may quickly reconnect to another cell.

[22] SigOver

Adaptive Overshadowing (AdaptOver)
Idea: Overshadowing the legitimate response to a NAS Service Request issuing a NAS Service Reject, which results in the UE not trying to connect again for $12-20$ hours.

Analysis: Low jamming-to-signal ratio (J/S), thus stealthy (not as obvious as a fake BS). Only downlink affected, thus undetected by the base station.

LTrack — IMSI Extractor
Idea: Overshadowing the legitimate response to a NAS Service Request issuing a NAS Identity Request, and sniffing the uplink, which results in the attacker learning the IMSI of the victim UE.

LTrack — Passive variant
This attack allows a fully passive localization of the victim. On the Downlink, Base Station notifies UE about the propagation delay between them, specifies a ring around a base station and travels unencrypted on MAC layer. In LTE-A, UE connects to multiple base stations. On the Uplink, Reference Signals used for channel correction. Propagation delay is channel condition. Observe propagation delay from multiple points. More accurate than 2G and 3G since 4G requires tighter synchronization.

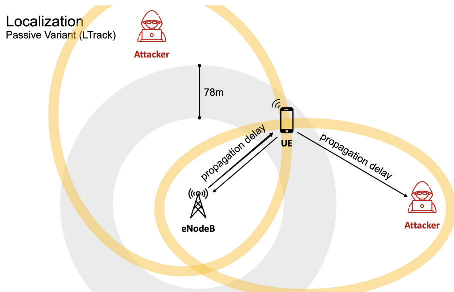

Figure 48: LTrack Passive Localization

4G Summary New crypto algorithms, new core network. Small security improvements (key hierarchy, handover protection), but not yet perfect.
Types of attacks: SigOver, fake base stations, man-in-the-middle.
Attack properties: stealthiness/detectability, power requirement, J/S ratio.

9.6. 5G

Overview Currently being deployed (2019/2020).
Radio link: *5G New Radio NR*, optimised OFDM, massive MIMO, two frequency ranges (FR1: sub-6GHz, FR2: mmWave range, 24-100GHz, high-throughput, high-bandwidth). Beam management to steer beams with a phase array allows connecting more devices.
Time Division Duplex TDD allows the same channel/frequency to be used for both up- and downlink, with different time intervals for different directions.[23] On one hand this allows flexible allocation, but on the other it requires precise synchronisation!

[23]Compare this with LTE which used FDD: the uplink and downlink used different frequencies.

Attacks Some ideas as research is ongoing.

- Beam stealing: attack beam training to steer beams away from victims (shown for IEEE 802.11ad)
- Broadband jamming (DoS): increasingly difficult due to large bandwidth (power constraint) $\implies$ need protocol-aware spoofing for DoS (challenge: tight synchronisation).
- PSS[24] spoofing: soft takeover: synchronise to cell, introduce PSS at correct timing then slowly move peak away (see GNSS subsection 4.1).

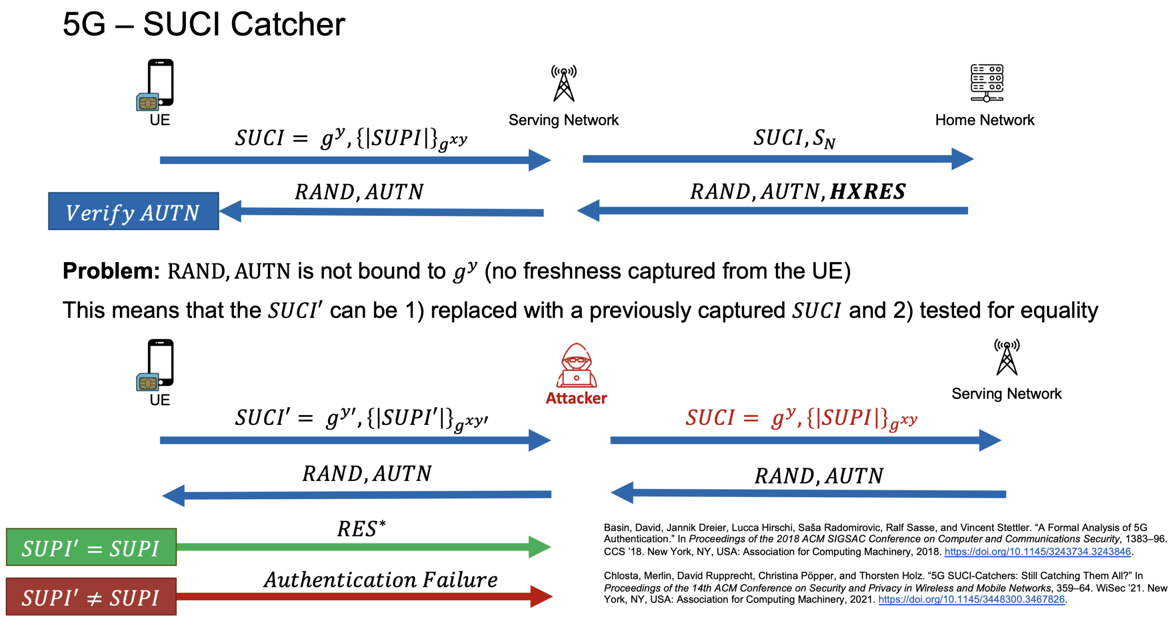

Figure 49: SUCI Catcher

5G Security Summary Similar crypto algorithms. Better replay protection for AKA (SIM generates nonces). User tracking mitigations (SIM can encrypt IMSI/TMSI with home operator's public key, stricter policies for changing temporary ids).

9.7. Questions

What capabilities does an adversary need to perform location tracking on a 4G network? He needs to get a hold of the IMSI of the UE. This can be done overshadowing a legitimate message after a service or attach request with a Identity Request message, to which the UE will happily reply with its IMSI. Or, the attacker could use two passive base stations to record the propagation delay and use triangulation to calculate the origin of the signal.

Describe the security requirements (Authentication, Encryption and Integrity) for the control and user plane in 4G. What does the standard prescribe/recommend? Do the operators follow those recommendations? MISSING DESCRIPTION. Authentication is required in both the control and user plane, encryption is optional for both but often used, integrity is mandatory for control plane but not user data.

[24] Primary Synchronisation Signal

	1G	2G	3G	4G	5G
crypto algorithms	none	weak	strong	strong	strong
AKA	none	one-way	mutual	mutual	mutual
core network	SS7	SS7	SS7	EPC	EPC
tracking	easy	limited	limited	limited	more limited?
fake BS	easy	easy	slightly harder	becoming feasible	challenging?
jamming / DoS	possible	possible	possible	possible	more challenging

Figure 50: Cellular Security Summary

10. Security in Critical Transport Infrastructures

Air Traffic Control ATC

- **Primary Surveillance Radar PSR**: ground based, measures time delta between transmission and reflection ⇒ independent.
- **Secondary Surveillance Radar SSR**: Transponder based interrogation ⇒ dependent. Mode A (identification code), Mode C (identification code + barometric altitude), Mode S (selective addressing to interrogate a specific aircraft, used in *Traffic Alert and Collision Avoidance System*)
- **Automatic Dependent Surveillance-Broadcast ADS-B**: Aircraft determines its position via satellite and regularly broadcasts the result. Replaces functions of SSR, and enables inter-aircraft situational awareness.

Problem Statement Huge number of systems and protocols in aviation (see Figure 51). None has confidentiality, integrity or authentication. On the other hand, attacker capabilities grow as domain knowledge spreads and software defined radios become cheaper.
At the same time change is incredibly slow due to certifications and legacy compatibility.

10.1. Privacy Issues in Aviation

No confidentiality of aircraft-ground communication Clear-text transmission of passenger medical status, forgotten belongings, credit card transaction details, etc.

Proprietary crypto ACARS[25] datalink encryption uses a mono-alphabetic substitution cipher with a limited keyset hardcoded in a lot of private/military/government jets.

[25] Aircraft Communications Addressing and Reporting System

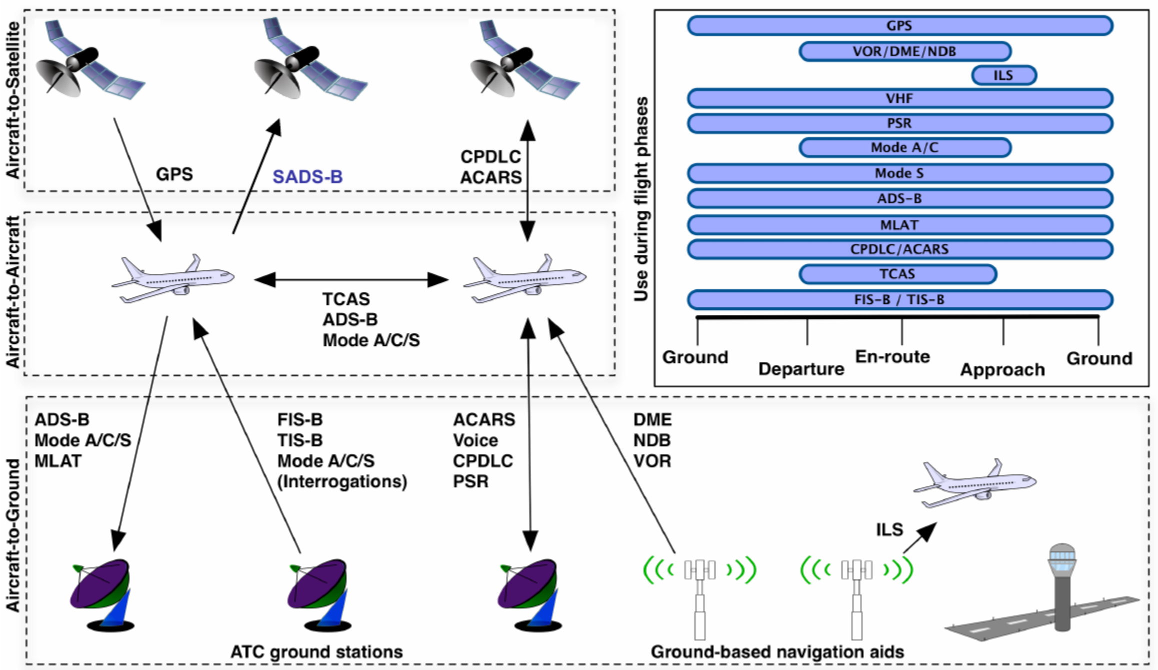

Figure 51: Overview of aviation systems

Aircraft identifiers Aircraft transponders have unique IDs that include aircraft type and operator. These are not easily and legally changed (unless aircraft is sold).

No location privacy Aircraft are globally trackable by anyone. Either using websites like Flight-Radar24 (heavily filtered), ADS-B Exchange, OpenSky Network. Or collecting own data from SSR and ADS-B signals using a cheap radio.

Possible "uses" for tracking: government aircraft movement, mergers & acquisitions (M&A) activities.

Mitigations: Block aircrafts on tracking websites, obscure ownership (register to shell/trust companies), disable position broadcasts (still easily localized near departure/destination airports), use commercial transport.

10.2. Security Issues in Aviation

Attacks The usual candidates: jamming, modification, injection (ghost aircraft = DoS on ATC).

Safety vs Security *Safety* is about dealing with accidents and failures. We tackle it with experience (root cause analysis) and redundancy (decreasing the likelihood of failure of the entire system).

Security on the other hand is concerned with protecting against an intelligent, adaptive attacker. See the Swiss Cheese model in Figure 52.

Traffic Collision Avoidance System TCAS Aircraft continuously predict each others' paths. If the paths are too close but not yet at risk, a *traffic advisory TA* is issued (and announced in the cockpit). If they remain on a close path, a *resolution advisory RA* is announced as compulsory instructions.

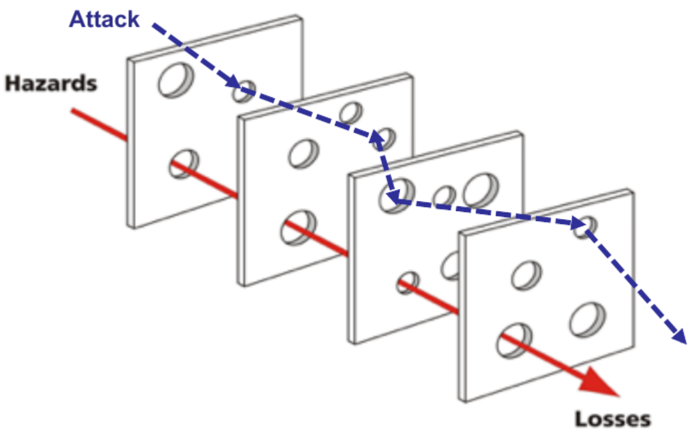

Figure 52: Swiss Cheese Model: Safety vs Security

Attack: Attacker listens to target aircraft and injects TCAS responses, forcing a TA and RA, thus forcing the aircraft to make unwanted course changes (plus pushes the pilots to reduce TCAS sensitivity).

10.3. Short-term Security Countermeasures

Cyber-Physical Countermeasures There won't be any real crypto any time soon. In the meantime, exploit physical layer data (timing, signal strength, Doppler shift, etc) to validate signals. Throw in some machine learning as well (for anomaly detection).
This hopefully lifts the bar back up to nation-state-attackers-only,

Multilateration Use multiple ground stations to receive (and validate) signals from aircraft.

OpenSky Network OpenSky Network is a crowd-sourced ATC system. Volunteers install software-defined radios around the world to capture signals and publish them.

Short-term solution: rather than cryptographically authenticating messages between aircraft, use ground network to check if everybody else also received the same signal.

Advantages: Does not touch legacy systems, low cost, global coverage, flexible.

10.4. Satellite and Maritime Infrastructures

Satellite links Signals are receptable in a large area (continent-scale). This allows tracking of e.g. military aircraft from far away.

Maritime VSAT (Very Small Aperture Terminal). Still large and expensive. Connects ships to IP network on land (WiFi, fleet monitoring, weather, navigation, cargo, etc). Composed of a satellite uplink (large beam, since satellite needs to cover a large area ⇒ can be captured from far away) and a directed downlink (towards a ground/land station).
Example: *Electronic Chart Display and Information System ECDIS* (paper chart/map replacement) receives updates via VSAT.

Analysis: Lots of interesting yet unencrypted traffic – from standard DNS/VoIP/IMAP to specific "ship data".

TLDR It's bad.

11. Bluetooth, Classic and Low Energy — BTC, BLE

11.1. Overview

Bluetooth Classic and Bluetooth Low Energy are similar technologies used for communicating short occasional messages. The former is used, for example, to stream music from a smartphones to wireless headphones; the latter is used with fitness trackers, smart sensors, . . . It can communicate over quite a long distance ($\sim 100m$).

11.1.1. Physical Layer

Bluetooth operates in the $2.4GHz$ band, spanning $80MHz$. For the modulation, it uses Gaussian Frequency Shift Keying (GSFK) and the bit-rate ranges from $125\,kbps$ to $2\,Mbps$. It uses 40 channels with $2\,MHz$ spacing, with 3 advertising channels and 37 data channels.

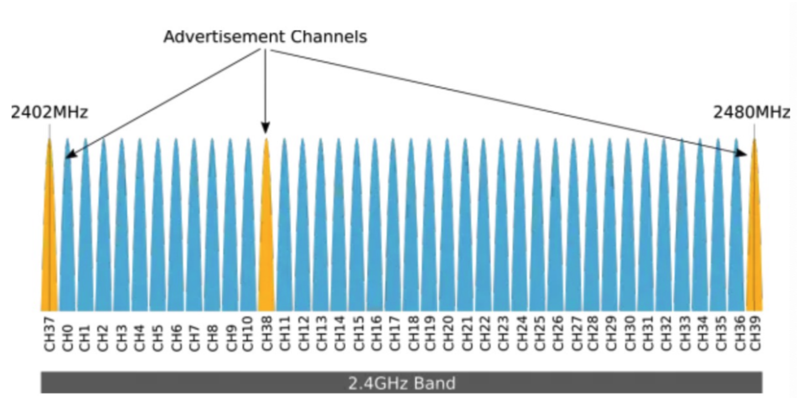

Figure 53: BLE Communication Band

In order to mitigate potential jamming and interferences, BLE uses frequency hopping, where each packet is transmitted over a new channel and whose schedule is negotiated during connection establishment.

11.2. Digital Contact Tracing

Goals *Complement* manual contact tracing an a pandemic. Notify users that they have been exposed to a person that tested positive (close than 2m for longer than 15min). In a timely, scalable, cross-border, yet secure and privacy-preserving manner. Using existing technologies and existing devices.

Exposure Notification Deployed by Apple and Google. Inspired by DP-3T. Used by many countries.

Idea: Phones broadcast ephemeral BLE[26] beacons. Neighbours record beacons. When infected, phone uploads the beacons to a public board. Phones poll the public board.

DP-3T See the white paper here.

[26]Bluetooth Low Energy

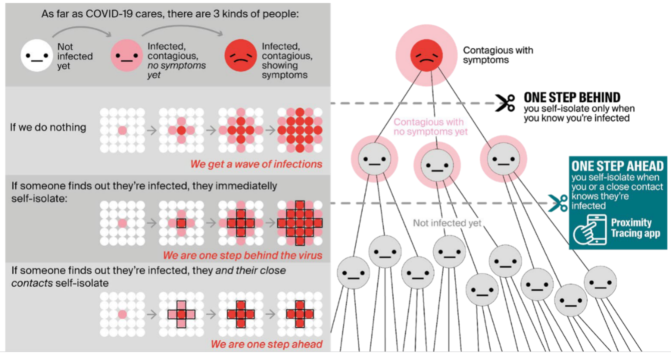

Figure 54: Contact Tracing: Motivation

Open questions More evidence is needed in some of the following areas:

- How reliable is Bluetooth at estimating proximity?
- How reliable are users to respond to a notification or trigger one?
- What is the financial impact of self-isolating in response to a notification? What demographics are more likely to comply?
- Access to smartphones, digital exclusion of certain groups.
- Is digital contact tracing a decisive factor?
- Inherent security[27] and privacy issues[28]
- Public communication, building trust.

[27] Relay+replay attacks, etc.

[28] E.g. if you only met a single person in two weeks and receive a notification, you can reliably de-anonymise them. However, contrast digital (privacy preserving) contact tracing with the data accumulated in manual contact tracing.

A. Imprint

This document closely follows the lecture slides of the *Security of Wireless Networks* lecture in the autumn semester 2020 at ETH Zurich. Our contribution to this is editing the whole lot and refactoring even more so that it may fit the "lecture summary" style. However, basically all graphics are copy & pasted from the slides. If you don't want yours here, please contact us and we will remove them.

In addition, this summary is based on a summary by Sarah Kamp.

An up-to-date version of this summary can be found at `https://github.com/eth-cs-student-summaries/Security-of-Wireless-Networks`. This version was built on December 18, 2022.