ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Security of Wireless Networks
Lecture 2/3

Srdjan Čapkun

Department of Computer Science

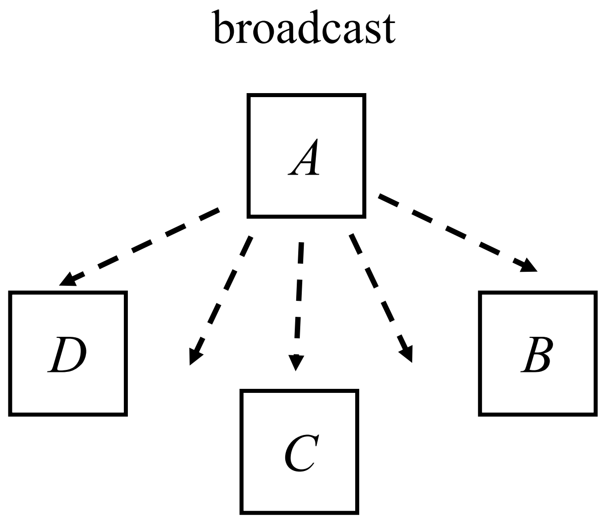
ETH Zurich

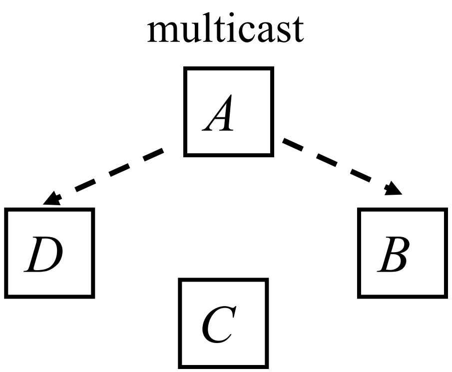
Broadcast Jamming-Resistant Communication
– keys, some keys, no keys –

Broadcast Communication

Broadcast communication

- One sender, many receivers
- Open system
  - New receivers may join, receivers may withdraw
  - Any receiver can listen (in contrast to multicast)





Examples:

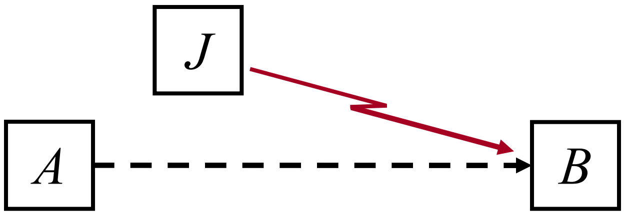
- radio (audio) broadcast (AM, FM, …)

- navigation signals: satellite-based (GPS), terrestrial (LORAN)

Attacks on Broadcast Communication

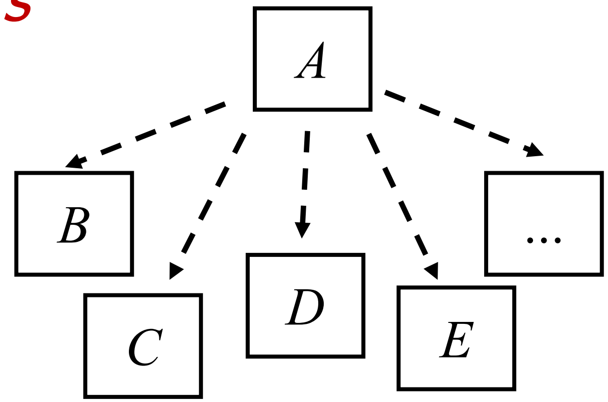
For pairwise (unicast) communication we only consider *external (outsider) attackers*

- A and B are mutually trusted
- Attacker uses only public information



Broadcast communication

- High and unknown number of receivers
- Receivers are potentially untrusted and may be colluding
- We need to consider *external attackers* and *internal (insider) attackers* (can be more efficient)
- Group keys?

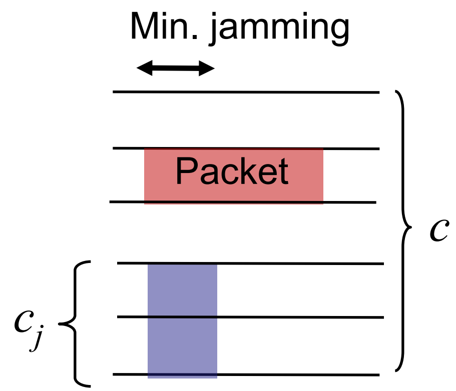


External Attackers on SS Techniques

External attacker

- Does *not* know the spreading code / hopping sequence
- Partial-band attacker can still jam. Example: FHSS

c = # frequency channels
c_j = # channels the jammer jams
n_j = # jamming cycles per packet (given by min. jamming period, packet length, and jammer capabilities)



$\Rightarrow$ p_j = Probability that the packet is jammed

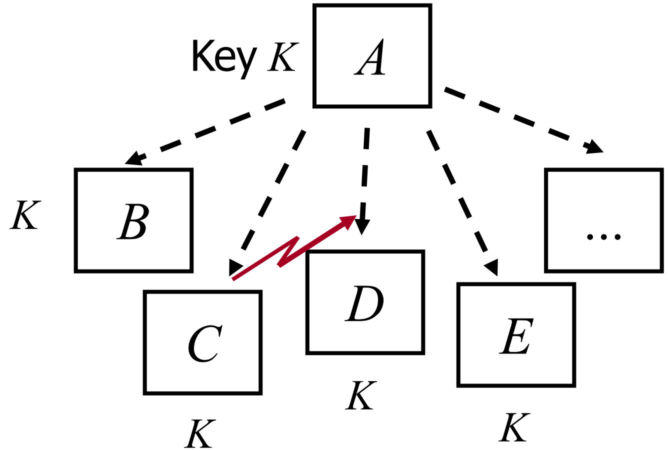
$$= 1 - (1 - c_j/c\,)^{nj}$$

Typical computation of jamming probability via the inverse

Internal Attackers on SS Techniques

Internal attacker

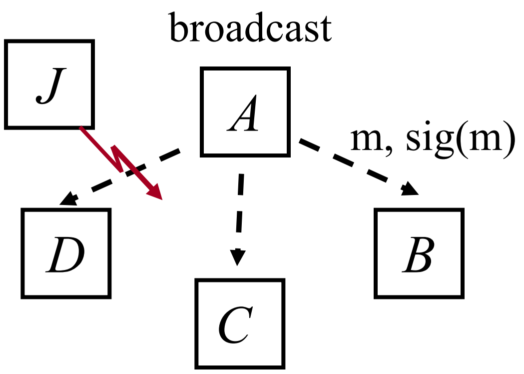
- Legitimate receiver: can decode the broadcast signal, i.e. knows the used spreading code and its synchronization
- Can *misuse the spreading code and synchronization* for jamming to disable other receivers to get the signal
- *Group keys do not prevent this attack!*
  We need a better solution!



Anti-jamming Broadcast

Side remark: Generally, anti-jamming (AJ) = jamming-resistant

Problem: Base station (BS) needs to *broadcast* an (authenticated / confidential) message to a *large number of receivers* in an *anti-jamming manner*



Desirable properties:

- Detect / prevent jamming
- Support a flexible number of receivers
- Tolerate a certain fraction of malicious receivers

Some solutions based on keys shared between sender and receivers:

1. Desmedt *et al.*: FHSS-based – each receiver listens to a subset of frequencies on which the sender transmits
2. Chiang, Hu: DSSS-based – codes assigned to each receiver

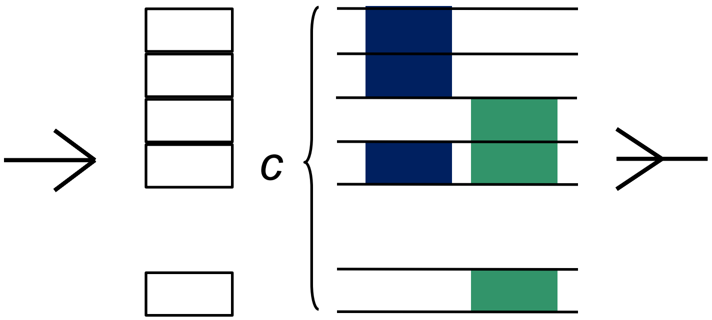
Broadcast Anti-jamming Systems [Desmedt et al.] - I

Broadcast anti-jamming based on frequency hopping (FHSS)

Coding method provides protection against malicious receivers

- Base station transmits the same signal simultaneously on multiple frequencies
- Each receiver listens to a subset of these frequencies at a given time
- *Threshold scheme*: provides protection against up to $j - 1$ colluding receivers

Based on secret information



Broadcast Anti-jamming Systems [Desmedt et al.] - II

Public Channel Allocation Table

- Defines the subset of channels where each receiver R_i is listening
- Known to every receiver
- $j-1$ receivers do not cover all channels of any other receiver
- Set coverage problem

Channel	BS	*R1*	*R2*	*R3*	*R4*	...
1	X		X			
2	X	X				
3	X	X		X		
4	X			X		
5	X	X			X	
6	X			X		
7	X		X			
8	X				X	
9	X		X	X		
...						

Secret Frequency Allocation Table

- The actual frequencies are secret
- Created and updated via a pseudo-noise generator

Channel	1	2	3	4	5	6	7	8	9	...
Frequency (in GHz)	2.437	2.462	2.417	2.442	2.447	2.457	2.412	2.422	2.432	...

[Snapshot of the frequency allocation table, the complete table is only known to the base station]

Broadcast Anti-jamming Systems [Desmedt et al.] - III

System Description:

- Channels $C = \{c_1, c_2, \ldots, c_m\}$
- Receivers $R = \{R_1, R_2, \ldots, R_l\}$
- Subsets of channels $CR = \{C_1, C_2, \ldots, C_l\}$

Theorem: If $|C_i| \geq 1 + (j-1)d$ for all $1 \leq i \leq l$ and $|C_i \cap C_k| \leq d$ for all $i \neq k$, then (C, CR) is a Broadcast Anti-Jamming System.

Sufficient but not necessary condition

Example: $C = \{1,2,3,4,5,6\}$, $R = \{B_1, B_2, B_3, B_4\}$,
$CR = \{\{1,2\}, \{2,3\}, \{4,5\}, \{5,6\}\}$

- Resistant to $j = 3$ jammers, i.e., $j - 1 = 2$
- $m = 6$, $l = 4$, $|C_i \cap C_k| \leq d = 1$
- Yet $|C_i| = 2$, not the required $|C_i| \geq 1 + (j-1)d = 3$

C	BS	B1	B2	B3	B4
1	X	X			
2	X	X	X		
3	X		X		
4	X			X	
5	X			X	X
6	X				X

Broadcast Anti-jamming Systems [Desmedt et al.] - IV

The Desmedt broadcast anti-jamming system works if

- the group of colluders consists of $j - 1$ or fewer members and hence each receiver is always left with at least one free (= unjammed) channel
- the assigned frequencies can be distributed over a broad, non-continuous frequency band

However, this scheme requires secret information to be shared between the base station and each participating receiver $\longrightarrow$ *multicast solution*

Dynamic Jamming Mitigation [Chiang and Hu] – I

Broadcast anti-jamming based on DSSS

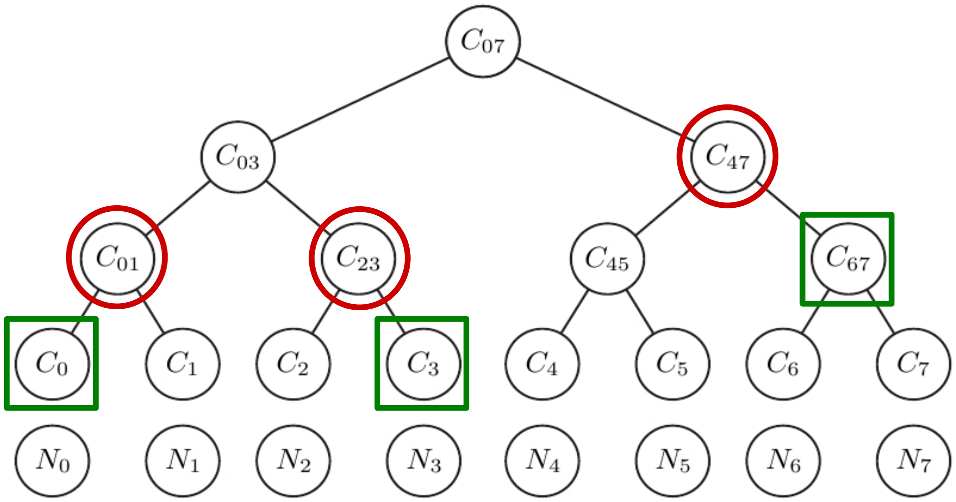
Counteract jamming by using a balanced binary key tree

- Each node corresponds to a spreading code
- Each user N_i is assigned to a leaf and knows all codes on the path from the root

The base station transmits on ...

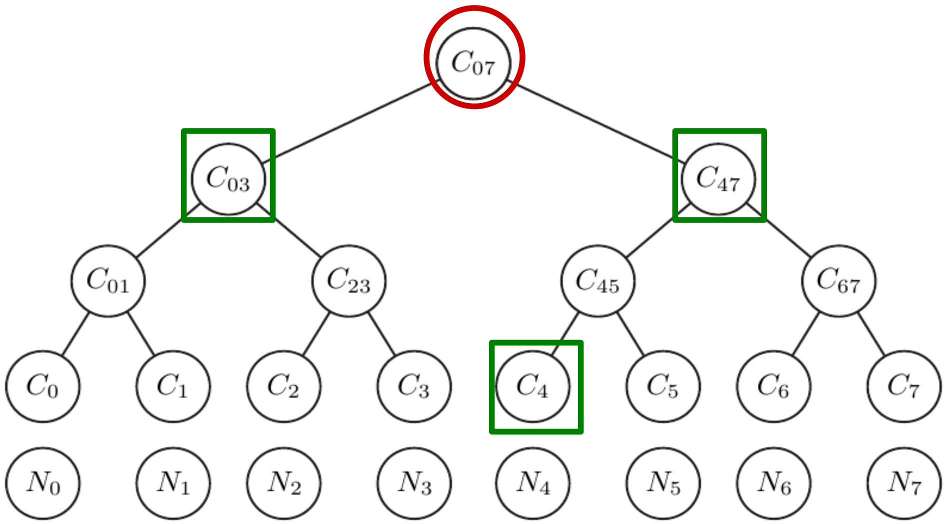
- a *disjoint cover of codes*, i.e., all users can decode using exactly one code
- a set of *test codes*

If a user receives a message on a test code but not on the corresponding detectable code, it reports jamming



Dynamic Jamming Mitigation [Chiang and Hu] – II

Jamming detection and mitigation



Detection and mitigation rely on feedback

Splitting and reforming the tree allows the transmitter to send each transmission on $\leq 2j+1$ codes, where j is the (expected upper) number of jammers (details omitted)

Dynamic Jamming Mitigation [Chiang and Hu] – III

Requires highly flexible base station (sending and receiving on a potentially large number of codes) and *feedback channels*

- Not applicable to unidirectional broadcast

Requires *secrets* to be shared between the base station and the receivers

- Each receiver knows the codes on its path to the root but no other codes
- Number of required secrets grows with the number of receivers

Looking back …

Introduction to broadcast systems

Group keys are not a solution against jamming

Two solutions based on secrets shared between the base station and the receivers:

- FH-based by Desmedt et. al
- DSSS-based by Chiang et. al

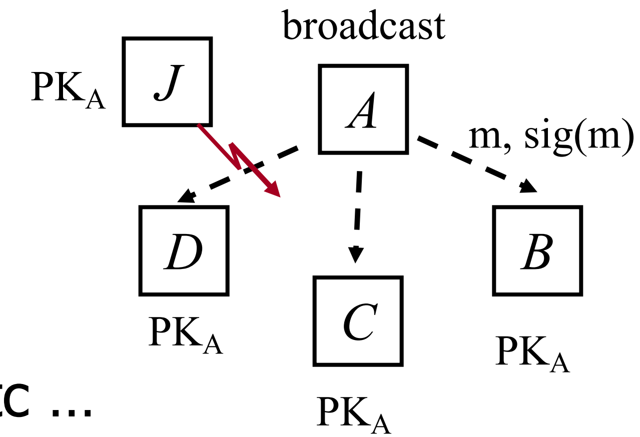
Can we achieve jamming-resistant communication without shared secrets?

Physical Layer Security

Broadcast Anti-Jamming Techniques
Without Shared Secrets

Anti-jamming Broadcast *Without Shared Keys*

Problem: BS needs to broadcast an (authenticated) message to a large number of *unknown/untrusted receivers* in an *anti-jamming manner*.

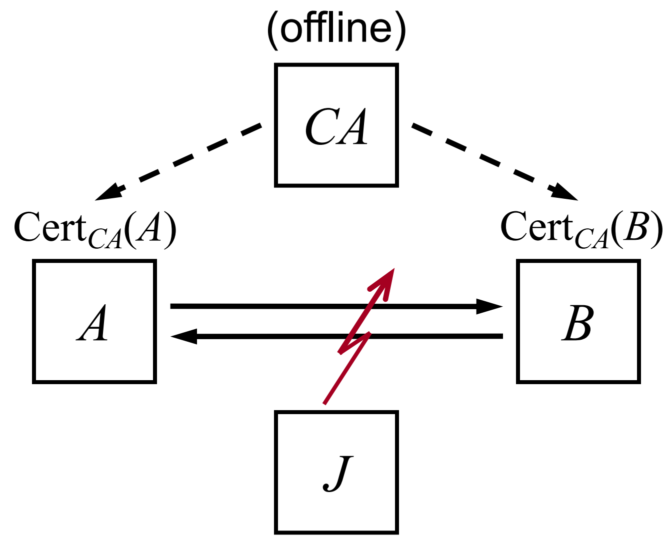


Applications: alarm broadcast, navigation signals, etc …

But …

- Anti-Jamming communication relies on shared secret keys
- In anti-jamming broadcast we cannot rely on shared keys (unknown/untrusted receivers)
- The prior schemes (Desmedt, Chiang) do not work for unknown receivers
- Public-key crypto does not help

Anti-Jamming Key Establishment



Problem:

A and *B* want to establish a shared secret key in the presence of a jammer *J*

Assumptions:

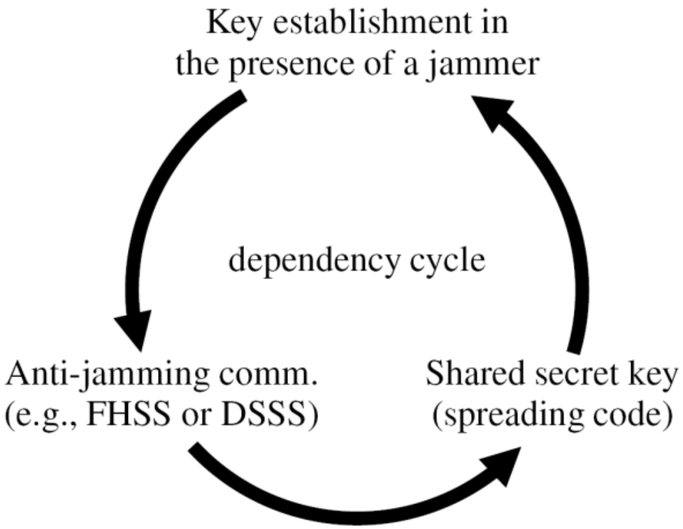
- *A* and *B* do not share any secrets
- The clocks of *A* and *B* are loosely synchronized O(s)
- Each node has a public/private key pair and a certificate binding its identity to the public key
- *CA* (Certification Authority) is trusted by all nodes; it may be off-line or unreachable by the nodes at the time of communication

Anti-Jamming / Key-establishment Dependency

Key establishment depends on jamming-resistant communication

Common anti-jamming techniques require a shared secret key (code)

Leads to an anti-jamming/ key-establishment dependency cycle



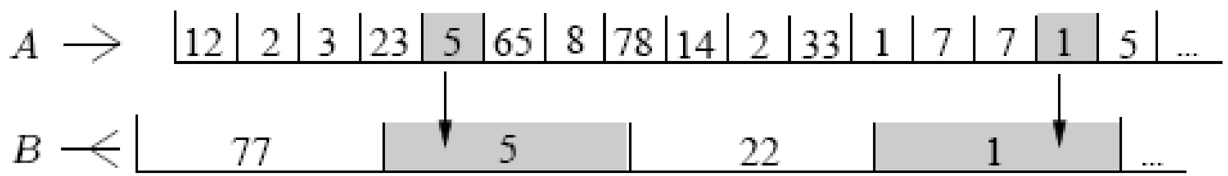
Two Solutions: UFH and UDSSS

Basic idea:

- If you cannot coordinate the sender and the receiver – Don't!
- Sender uses random hopping sequences / spreading codes unknown to the receiver (public set)

Two solutions:

- Uncoordinated Frequency Hopping Spread Spectrum (UFH)
- Uncoordinated Direct Sequence Spread Spectrum (UDSSS)



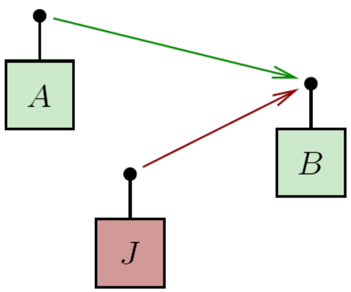
(UFH)

Rationale:

- The attacker cannot predict which channels will be used (neither can the receiver)
- Equivalent to FH in jamming protection (but not in throughput)
- Throughput can be improved by using broadband receivers (c_t, c_r)

Attacker Model

- Attacker goal: to prevent communication!
- Attacker actions: *Jam*, *Insert*, *Modify*
- Attacker types: Responsive, Sweep, Random, …
- Attacker strength (channels/time to jam/sense): c_s / t_s, c_j / t_j
- Power to insert, jam, and overshadow: P_t, P_j, and P_o



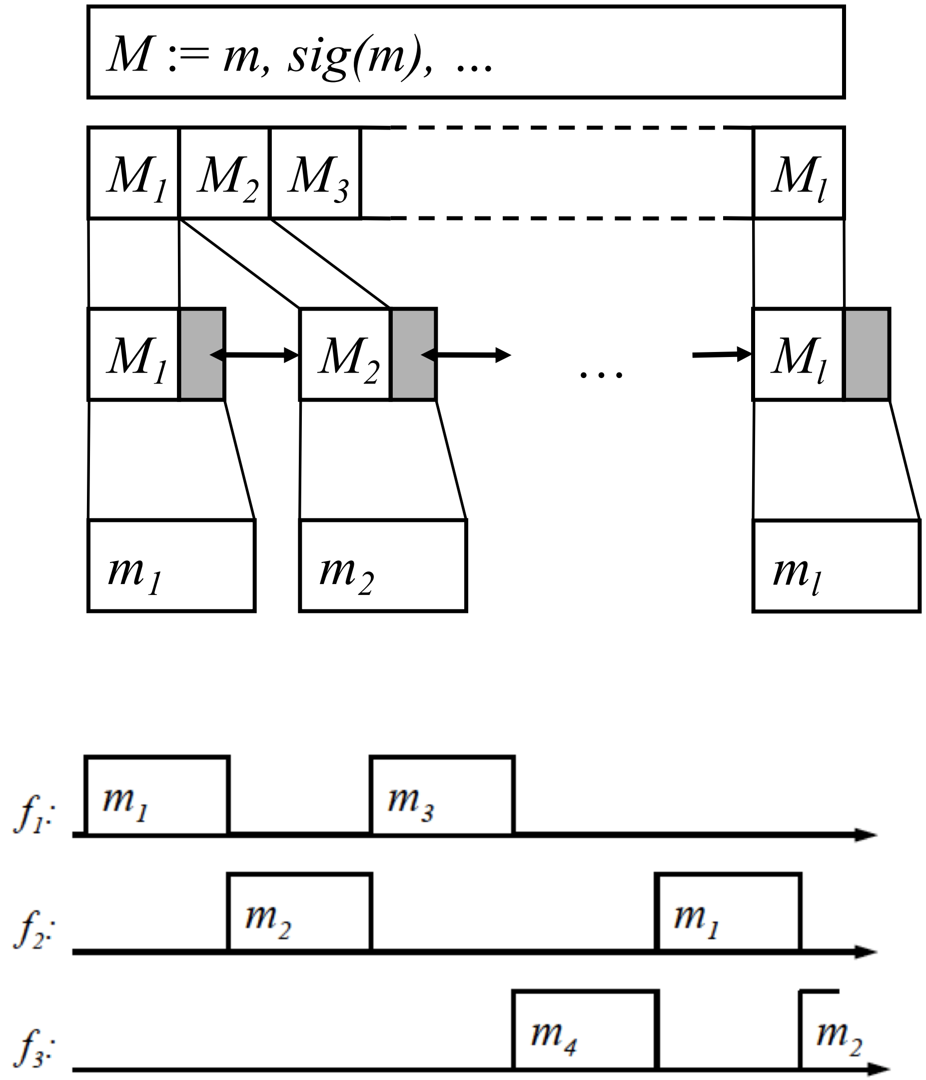
- P_T: total signal strength that attacker J can achieve at the receiver B

- Given the number of frequency channels on which the attacker inserts (c_t), jams (c_j), and overshadows (c_o),

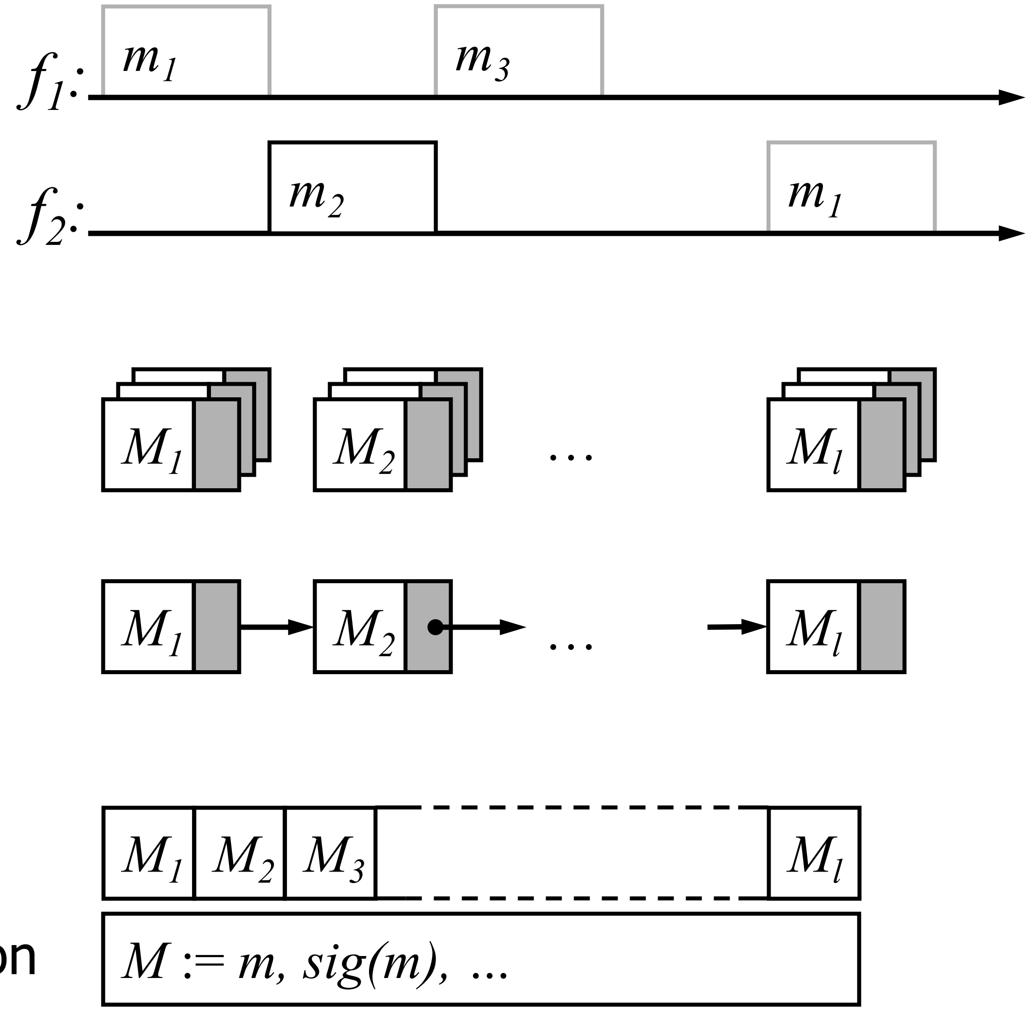
$$c_t P_t + c_j P_j + c_o P_o \leq P_T$$

Uncoordinated Frequency Hopping (transmitter)

1. Fragmentation
2. Fragment linking (protects against insertion)
3. Packet Encoding (ECC) (protects against jamming)
4. Repeated transmission



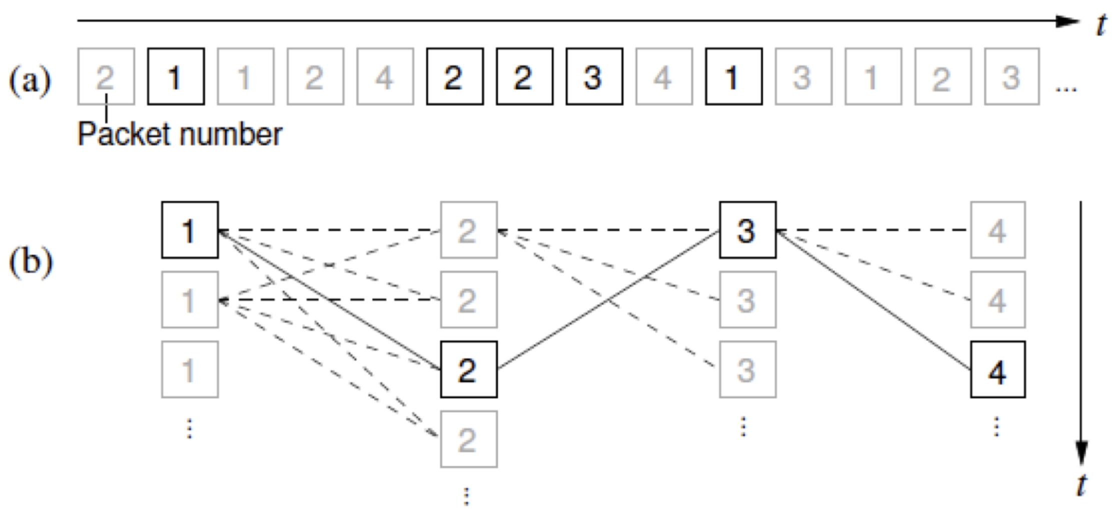
Uncoordinated Frequency Hopping (receiver)



1. Receiving packets
2. Packet decoding
3. Ordering and linking
4. Message reassembly and signature verification

Security analysis: Fragment linking

Problem: Fragments are not individually authenticated (pollution attacks)



$(I+1)^{\ell}$

Signature verification at each candidate message (after reassembly)

In the best case, I=1 … (depends on attacker's # of channels, power …)

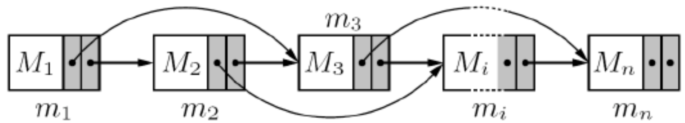
but ℓ is large; $\ell = \frac{\text{message size}}{\text{slot size}}$ (>20)

Result: Attacker performs a DoS attack on the logical level instead on the physical

Security analysis: Fragment linking

Problem: Fragments are not individually authenticated (pollution attacks)
Solution: Cryptographically link fragments (no reliance on shared key) to achieve message integrity

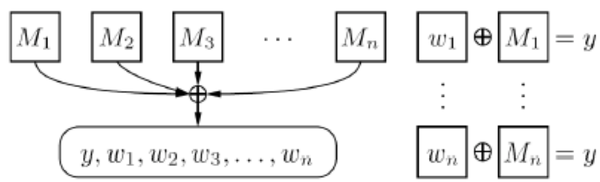
Hash linking



$$m_i := id||i||l||M_i||h_{i+1}||\ldots||h_{i+\alpha}$$

Min 1 hash

One-way Accumulators



$$m_i := id||i||l||M_i||w_i$$

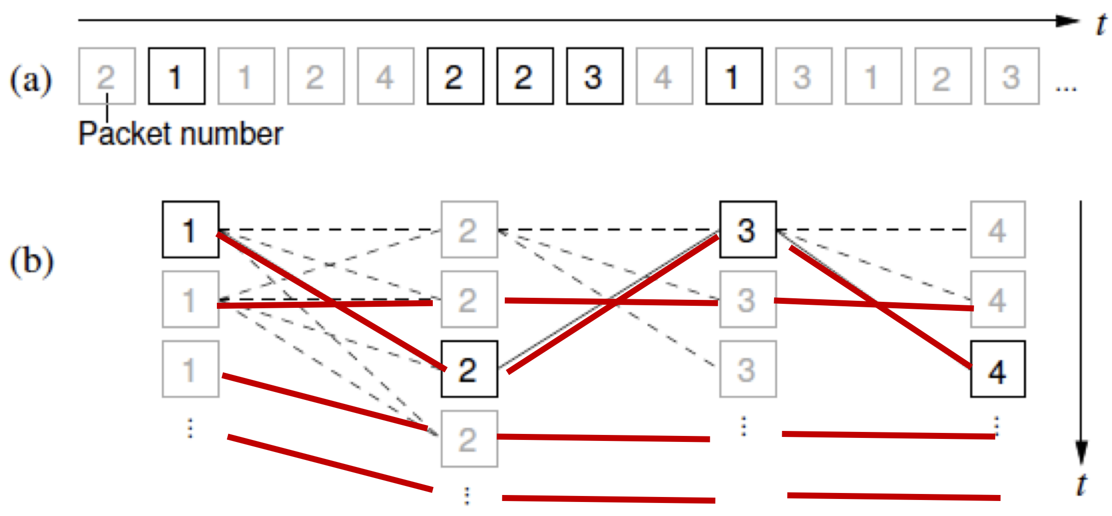
1 witness

Short signatures

$$m_i := K_M||i||l||M_i||Sig_{K_M^{-1}}(K_M||i||l||M_i)$$

1 short signature

Security analysis: Fragment linking



Gain: Instead of $(I+1)^{\ell}$ signature verifications, reduction to $(I+1)\ell$ hash/acum/signature verifications + $(I+1)$ signature verifications

Signatures and accumulators better than hash linking

Possible extensions:

- Use linking with erasure codes, e.g., Fountain codes.
- Reconstruct the message from any *k* fragments.

Security analysis: Packet Encoding

Defined by the jamming resistance ρ and coding rate r_c

- Packet transmission time:

$$t_m = |m|B/r_c$$

- #channels that the attacker can (blindly) jam during the transmission:

$$n_j := \frac{t_m}{\rho t_m + t_j}$$

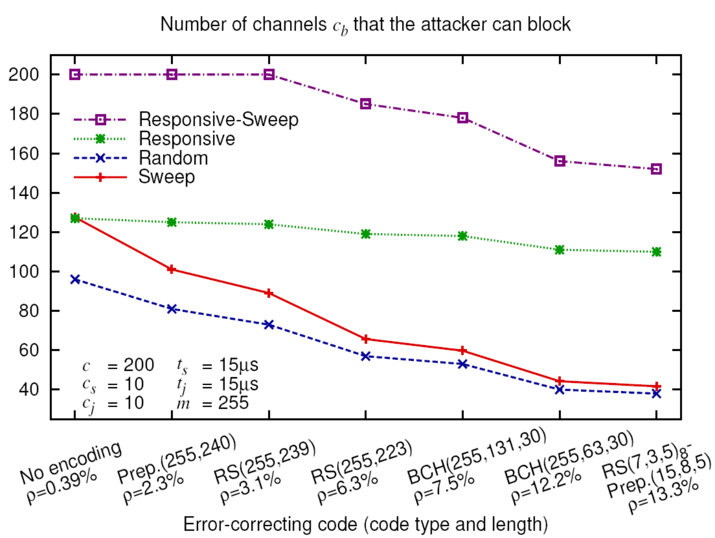
- #channels that the attacker can scan during the transmission:

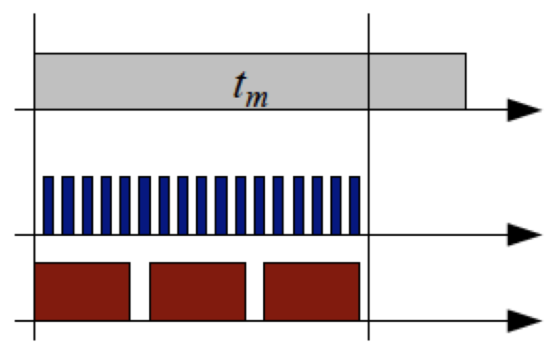
$$n_s := \frac{t_m - \rho t_m - t_j}{t_s}$$

- #channels that the attacker can block during the transmission

$$c_b = n_j c_j + n_s c_s$$

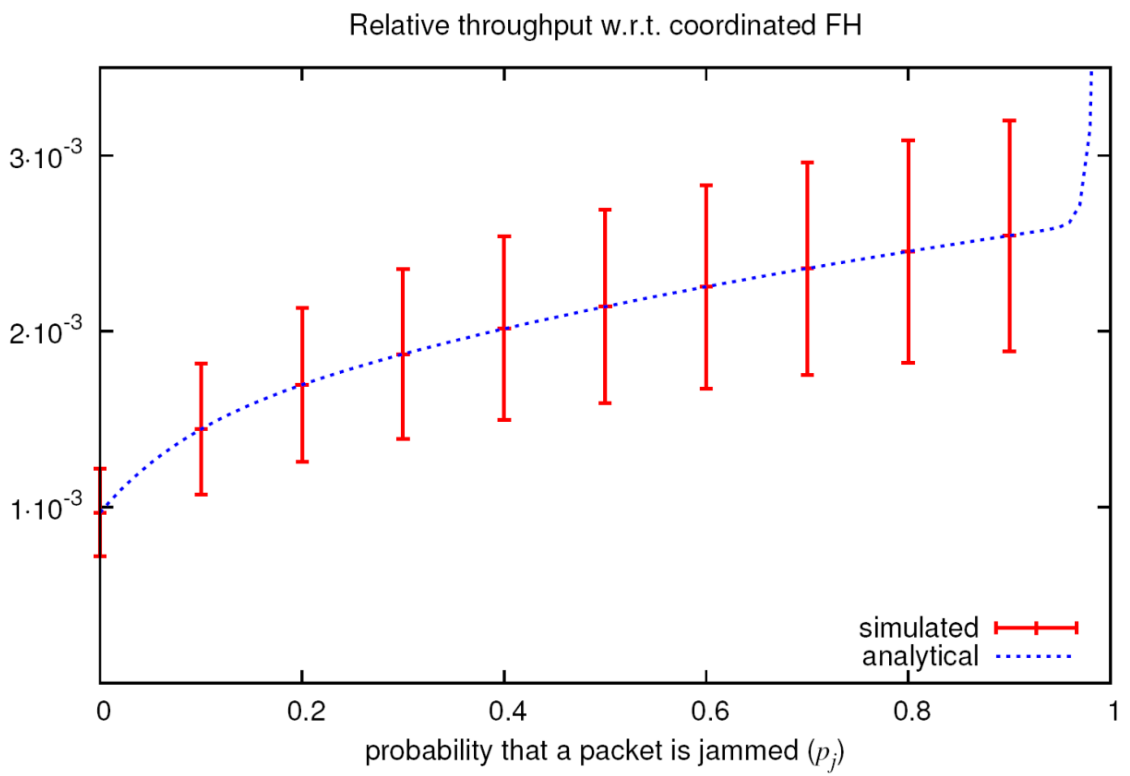
$$p_j = \frac{c_b}{c}$$





Performance Results

- Optimal # of channels ($c^*=2c_b$)



- Some results (c=200, 1MBit/s, 1600 hops/s, ECC signature, $|M=2176|$, ℓ=13)
  - Throughput: 1000x slower than FH
  - Latency: 2 – 100s (different attacker strengths)

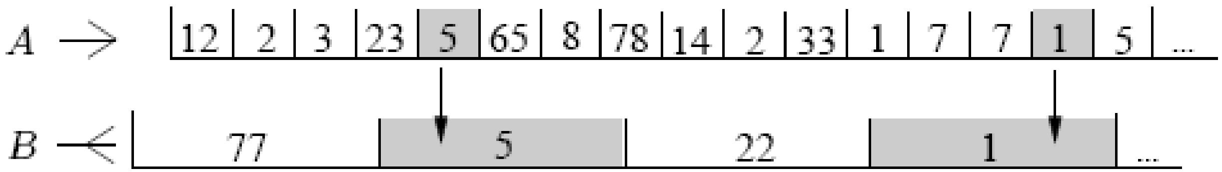
Two Solutions: UFH and UDSSS

Basic idea:

- If you cannot coordinate the sender and the receiver – Don't!
- Sender uses random hopping sequences / spreading codes unknown to the receiver (public set)

Two solutions:

- Uncoordinated Frequency Hopping Spread Spectrum (UFH)
- Uncoordinated Direct Sequence Spread Spectrum (UDSSS)



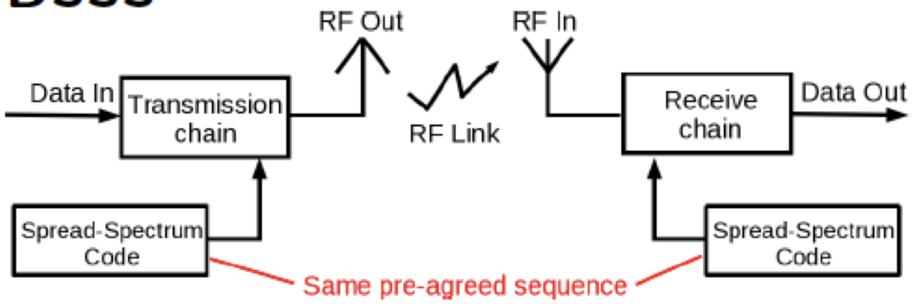
(UFH)

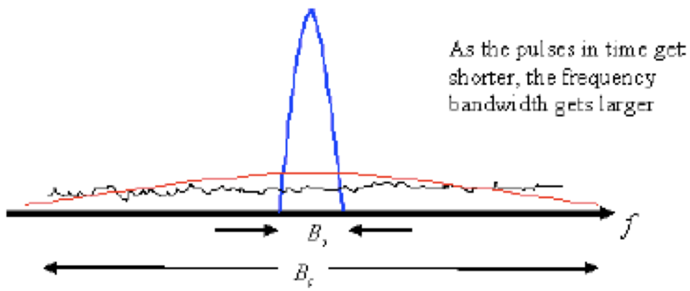
Rationale:

- The attacker cannot predict which spreading codes are used by the sender (neither can the receiver)
- UDSSS has reduced latency compared to DSSS
- Throughput can be improved by using parallelization

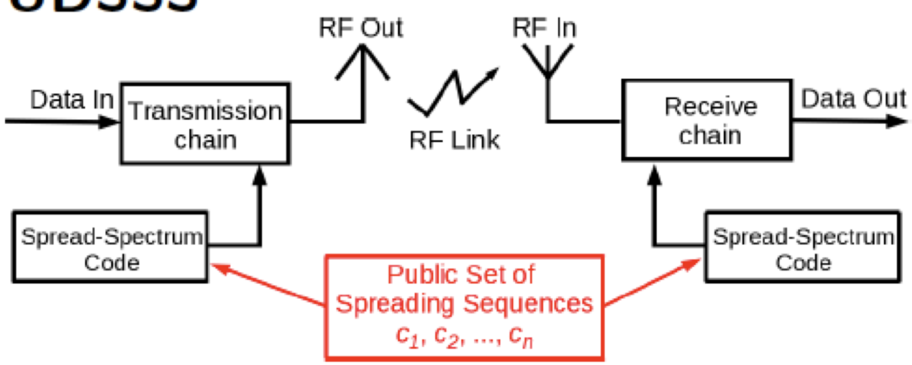
Uncoordinated Direct Sequence Spread Spectrum

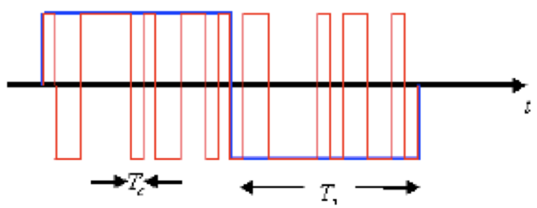
- **DSSS**





- **UDSSS**



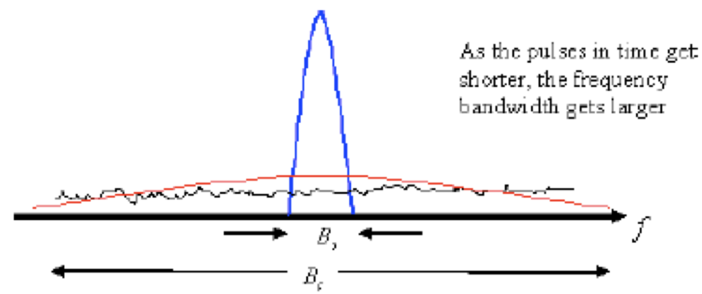


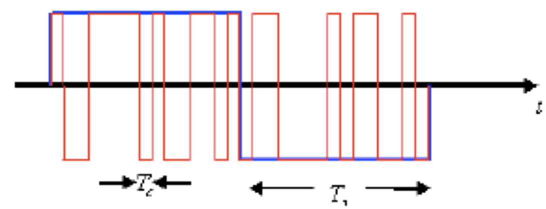
Uncoordinated Direct Sequence Spread Spectrum

- Public set C of spreading sequences

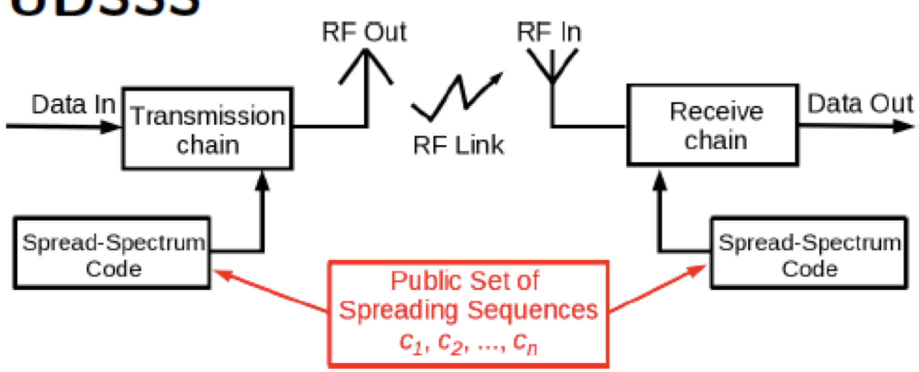
Sender randomly selects sequence $c_s \in C$ to spread message M

Receivers record signal and despread M by applying sequences from C using a trial-and-error method



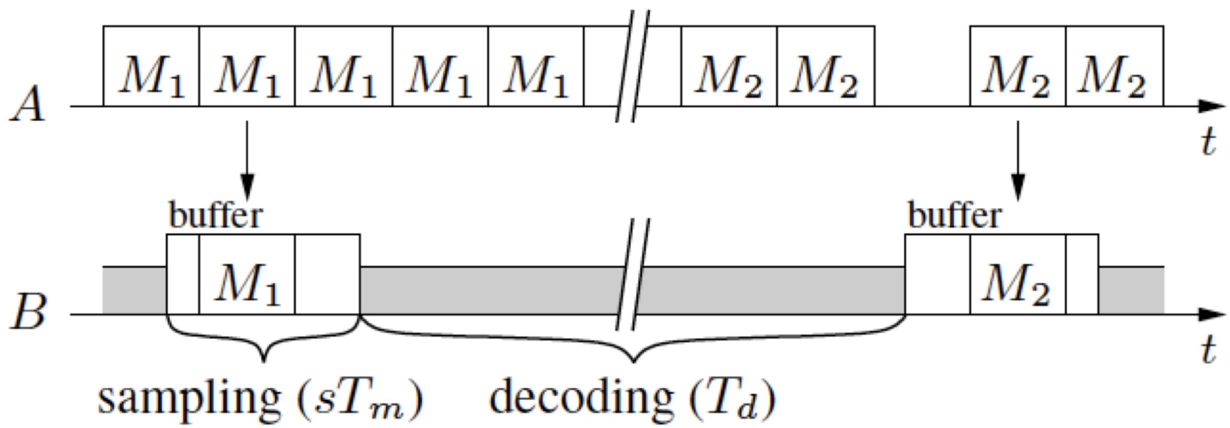


- **UDSSS**



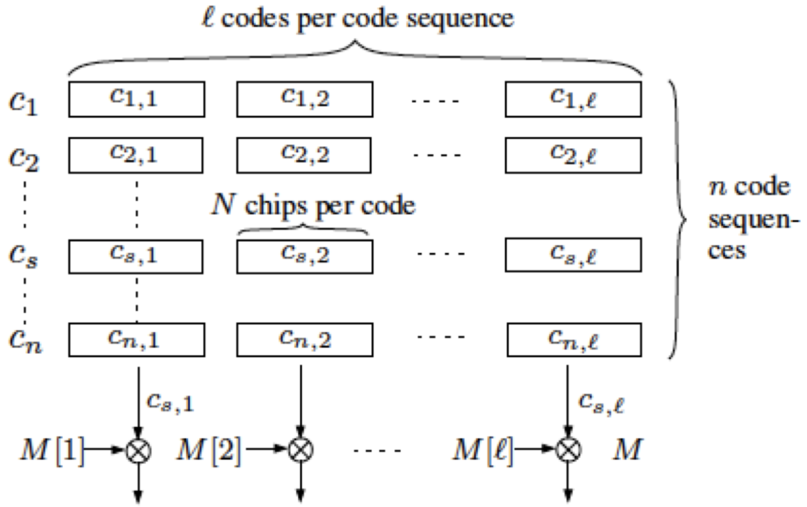
Uncoordinated Direct Sequence Spread Spectrum

- Message repetitions, due to
  - lacking synchronization between sender and receivers
  - the possibility of successful jamming attacks



Uncoordinated Direct Sequence Spread Spectrum

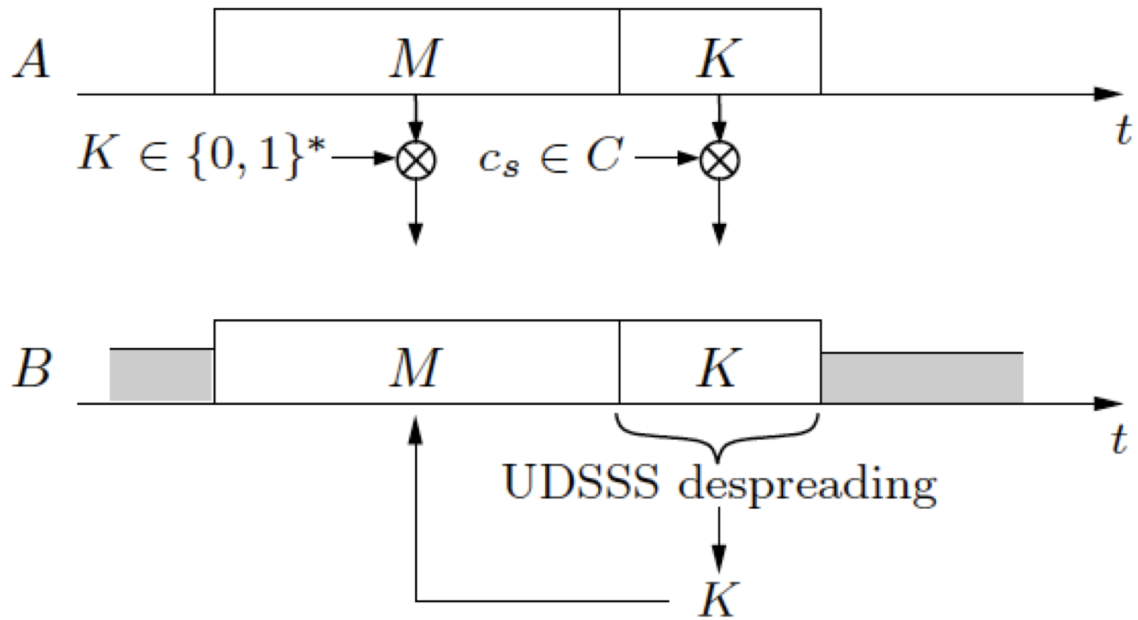
- Code set C composed of n code sequences
- Each code sequence is composed of ℓ spreading codes containing N chips
  - E.g., $N = 100$ chips $\rightarrow$ 20 dB processing gain
  - Auto-correlation and cross-correlation properties



- Successful despreading requires to hit the correct spreading sequence *and* the correct synchronization

UDSSS: Optimization

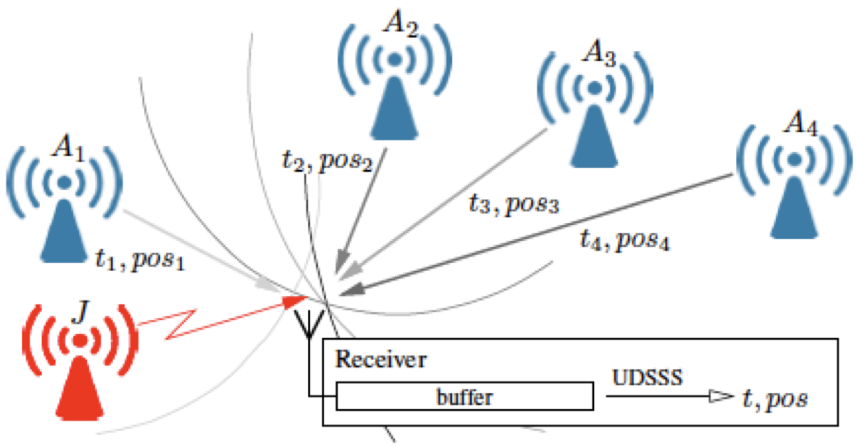
- **Idea**: Use UDSSS to transmit the spreading key only
- **Trick**: First transmit message M using a random spreading code K, then transmit the spreading code K using UDSSS

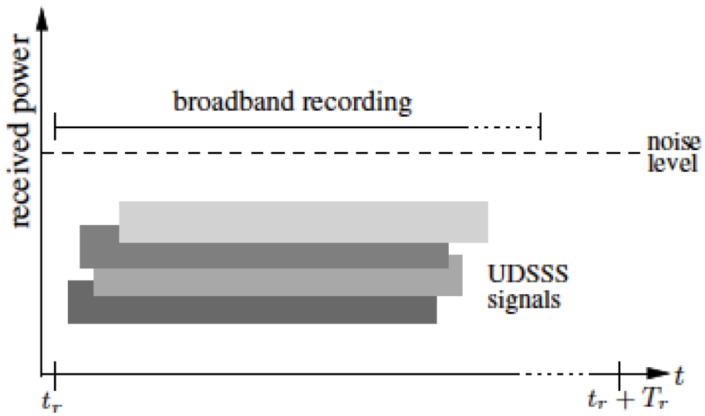


- **Advantages**: Smaller spreading code set. Quicker decoding. Longer messages. More flexible security level.

UDSSS: Example Application

- For positioning and/or time-synchronization
- Requirements:
  - signals from three to four different base stations
  - precise time-stamping of signal reception

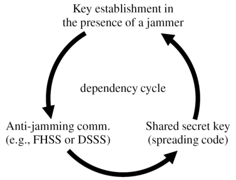




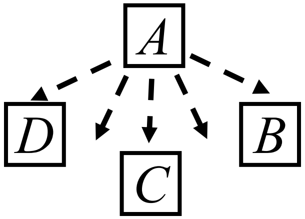
- UDSSS provides:
  - anti-jamming transmission of multiple signals in parallel
  - precise time-stamping of signal reception (despite delayed recovery) & updated time-stamps in each transmitted message
  - anti-spoofing protection of authenticated messages

Summary

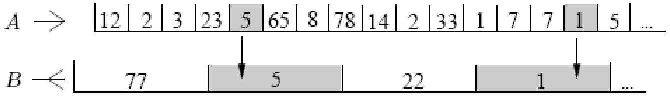
- Anti-jamming – key-establishment circular dependency



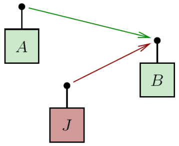
- Broadcast anti-jamming problem



- UDSSS and UFH



- New attacker models



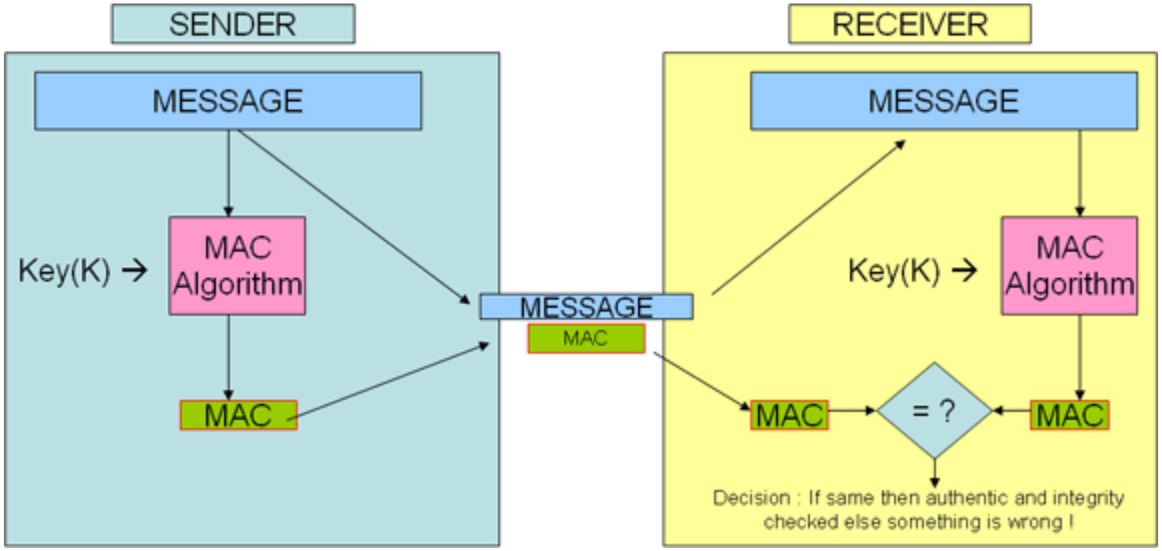
- Applications

Physical Layer Security

Application of (Broadcast)
Anti-Jamming Techniques to Key Establishment

Applications for Shared Keys in Wireless Networks

- Secret keys are required / used for:
  - Communication techniques (DSSS, Frequency Hopping)
  - Encryption of messages
  - Integrity protection of messages (MACs = Message Authentication Codes)



[www.answers.com]

  - Authentication / authorized access
  - …

The Problem with Key Establishment

Key establishment is a challenge

Pre-sharing Symmetric Keys

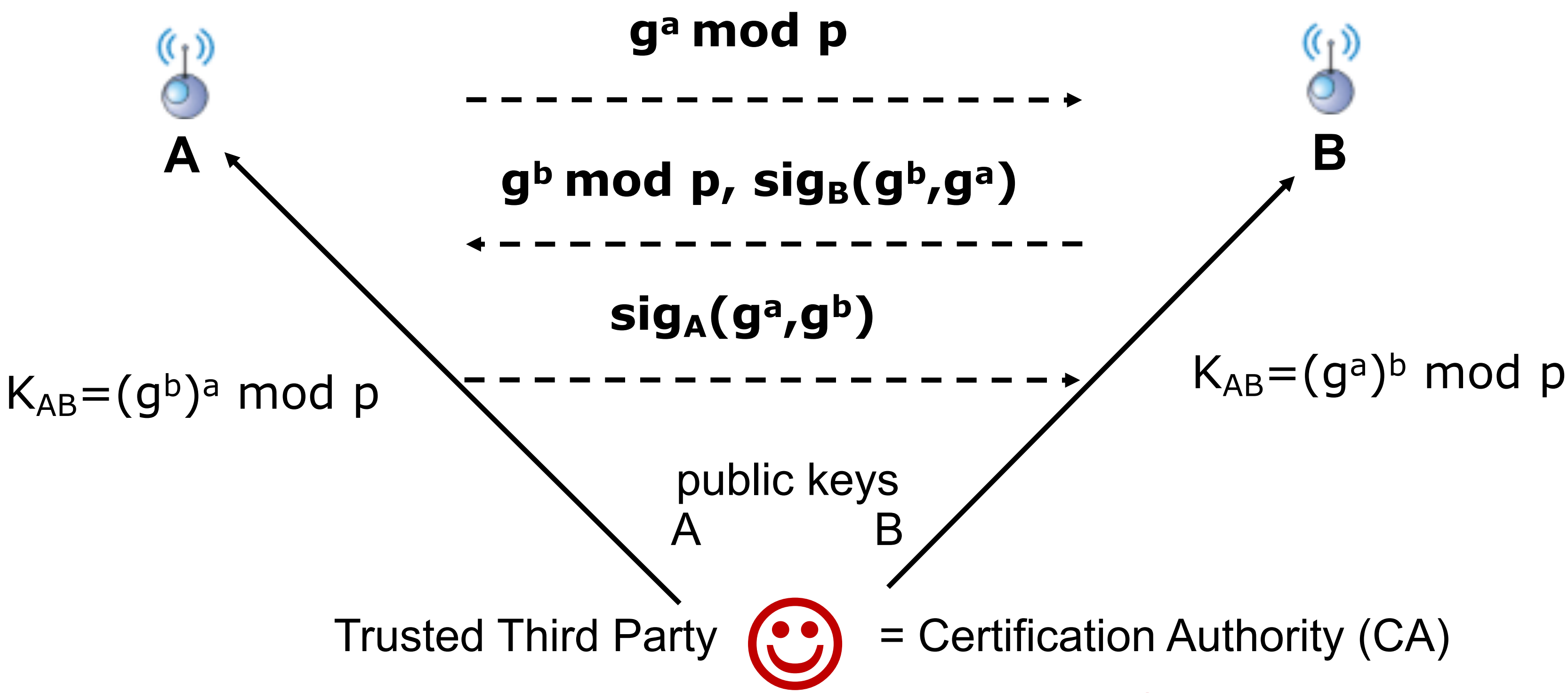
- A Trusted Third Party (TTP) pre-loads the keys
- Efficient (+)
- Suffers from network dynamics problems (–):
  - new nodes joining, key revocation, key compromise

Key Establishment

- Based on public-key (asymmetric) cryptography
- Prominent examples: RSA, Diffie-Hellman (DH)
  - Based on computational hardness of the factorization (RSA) or discrete logarithm (DH) problem
- Requires reliable communication

DH Key Establishment

- Nodes A and B do not share any secrets, but possess certificates of their public keys
- Authenticated Diffie-Hellman Protocol (using signatures)



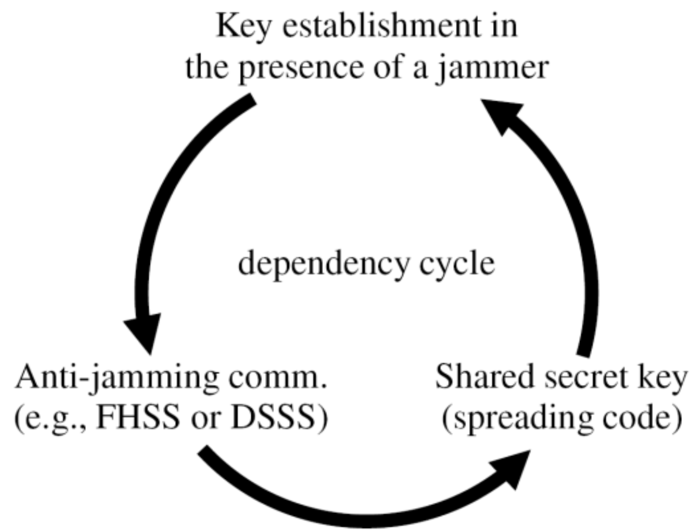
- *Conventional SS-Techniques cannot be used for the communication due to the missing shared secret*

Anti-jamming / Key-establishment dependency

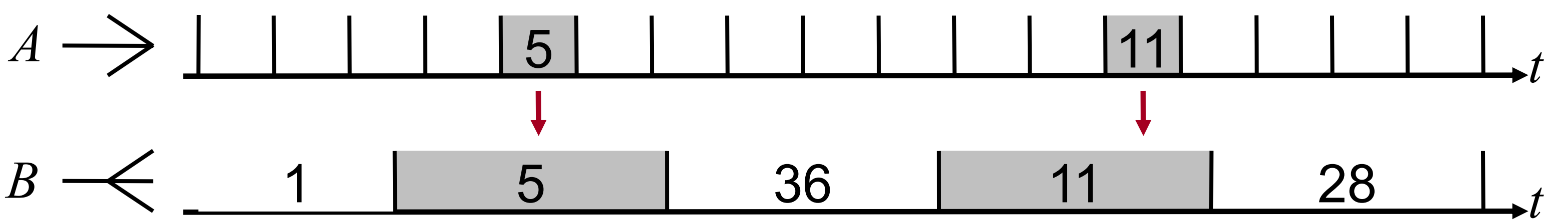
- Key establishment (e.g. using DH) depends on jamming-resistant communication

- Common anti-jamming techniques require a shared secret key (code)

- Leads to an anti-jamming/ key-establishment dependency cycle



- Key idea: break the dependency cycle by using **Uncoordinated Frequency Hopping** (UFH)



Key Establishment Protocol: Sender/Receiver

ECC-based Station-to-Station Diffie-Hellman

- P lies on elliptic curve $E(F_q)$, CA = Certification Authority
- PK_A = A's public key, Sig_A = A's signature, $r_A P$ = A's key contribution

A — **B**

$r_A \leftarrow_R Z$ — $r_B \leftarrow_R Z$

$A \rightarrow B$: $A, PK_A, Sig_{CA}(A, PK_A), T_A, r_A P, Sig_A(A, \ldots, r_A P)$

$K = r_B(r_A P)$

$B \rightarrow A$: $B, PK_B, Sig_{CA}(B, PK_B), T_A, r_B P, Sig_B(B, \ldots, r_B P)$

$K = r_A(r_B P)$

Elliptic Curve Cryptography (ECC) enables to reduce the key length while maintaining the level of security

- E.g., 128-bit security level [NIST] →256 bit prime fields on elliptic curves and 512 bit keys (vs. 3072-bit key for RSA)

Use UFH to transmit the messages

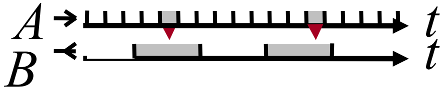
What to remember?

- What are broadcast systems?

- Applications for broadcast

- Approaches for enabling jamming-resistant broadcast despite internal attackers

- Jamming-resistant communication without shared secrets



- Anti-jamming/Key-establishment dependency