Jamming-resistant Broadcast Communication without Shared Keys

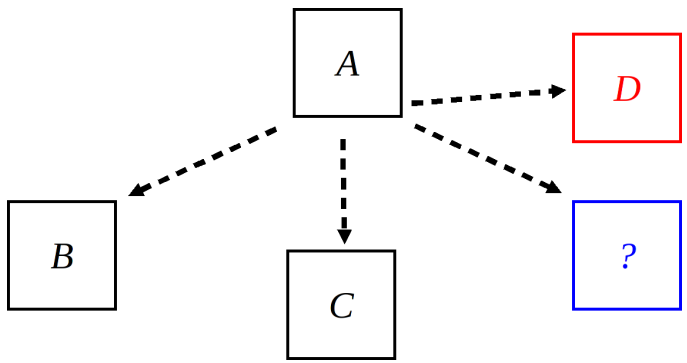
Christina Pöpper

Joint work with Mario Strasser and Srdjan Čapkun
System Security Group
ETH Zürich

August 2009

Broadcast Communication

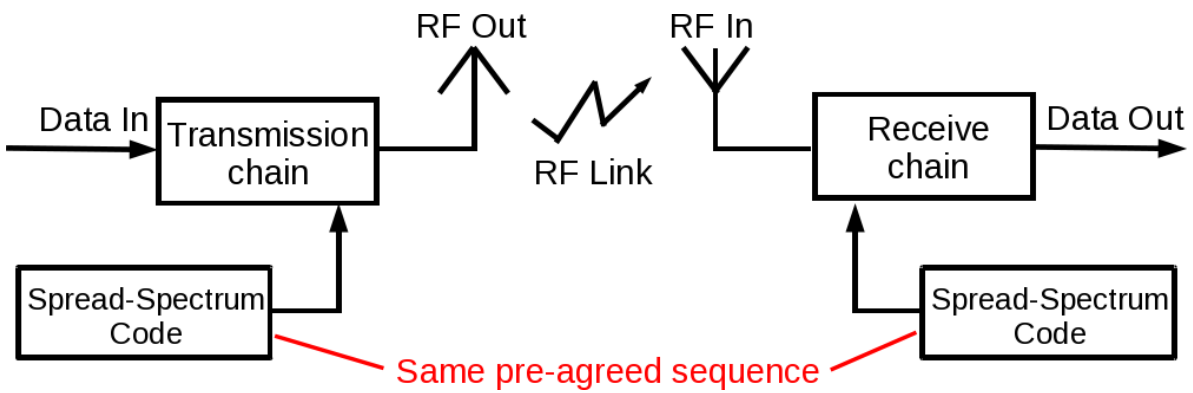
- **Setting**:
  - Broadcast of (authenticated) messages to a (large) number of receivers
  - Wireless RF communication
  - Receivers may be unknown and/or untrusted



- **Broadcast Applications**:
  - Alarm broadcast
  - Broadcast of navigation signals
  - ...

Jamming Attacks

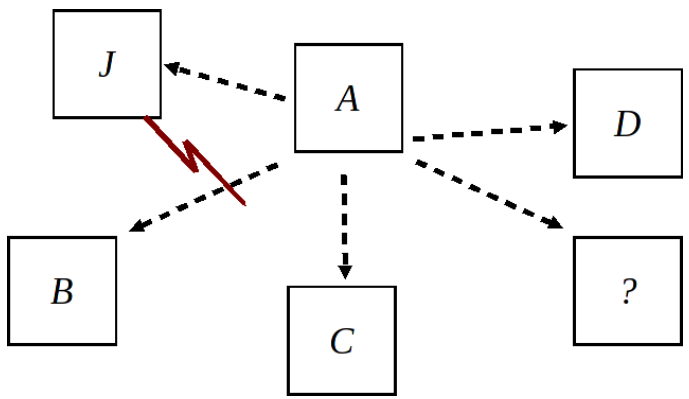
- **Jamming Attacks**:
  - Jamming devices are cheap and easy to obtain
- **Anti-Jamming Techniques**:
  - Spread Spectrum Techniques, e.g.,
    - Frequency Hopping Spread Spectrum
    - Direct-Sequence Spread-Spectrum (DSSS)



  - **Rely on a secret key (or code) pre-shared between sender and receivers before the communication**

Jamming Attacks

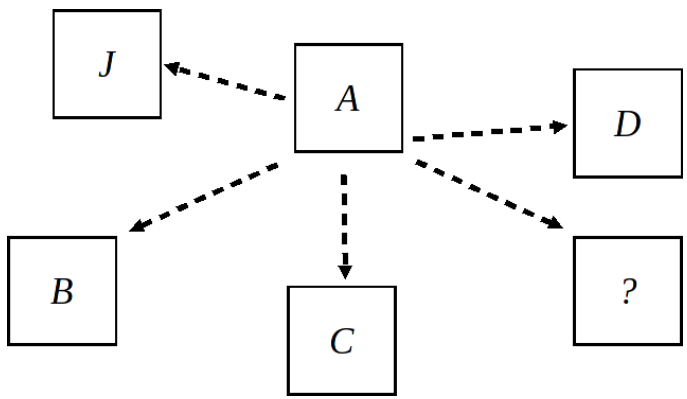
- **Anti-Jamming Techniques in Broadcast Settings**:
  - Pre-sharing keys is complex or infeasible
    - Public key cryptography does not help
  - Even if secret keys are pre-shared, receivers still need to be trusted



→ Anti-jamming Broadcast Problem

Problem Statement

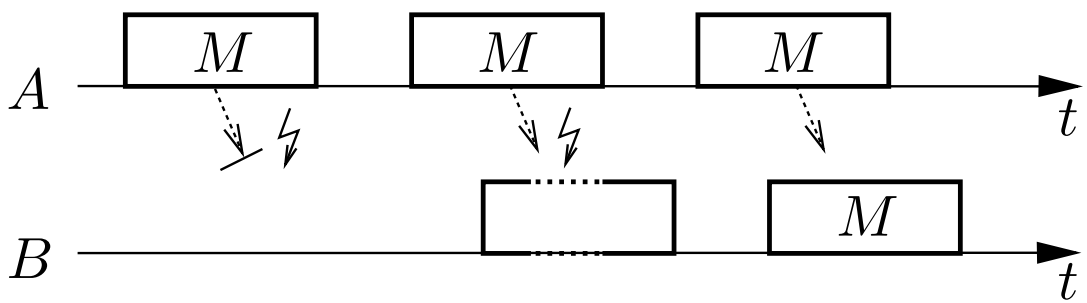
- **Problem Statement**: *How can we enable jamming-resistant broadcast communication if the sender does not share secret keys with (all the) receivers?*



- In [Desmedt et al., ICON99] and [Chiang et al., InfoCom08], solutions were proposed for jamming-resistant broadcast, but they rely on shared secret information

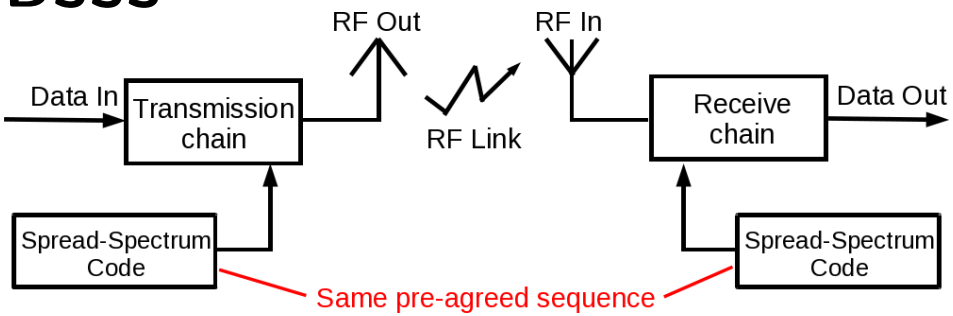
Our Solution

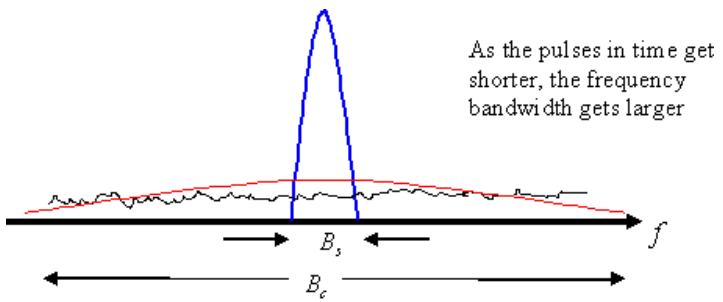
- **Anti-jamming Broadcast without Shared Secrets**
  - Scheme called Uncoordinated DSSS (UDSSS)
  - Achieve communication to an unknown/untrusted set of receivers in the presence of communication jamming
- **Key Idea**: Base the communication on DSSS but release the requirement of shared secret keys by randomization
- **Key Observation**: "Whatever has arrived unjammed at the receiver can be decoded"

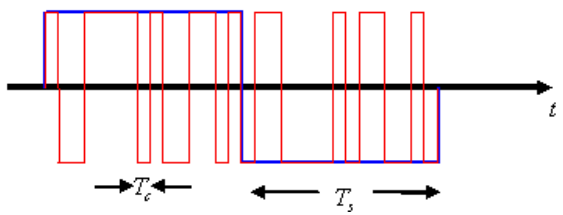


Uncoordinated DSSS (UDSSS)

- **DSSS**

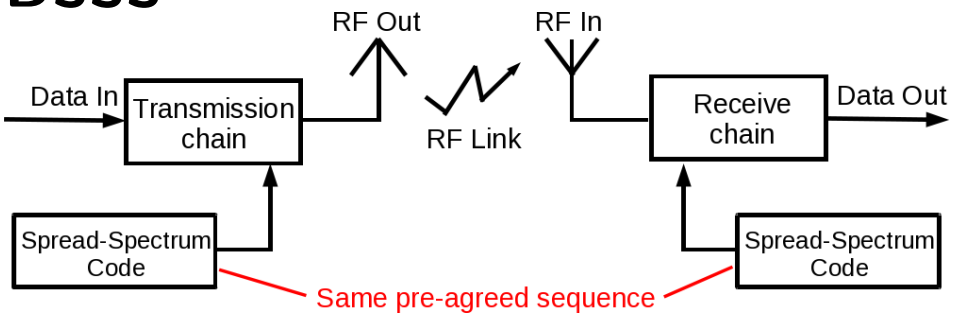


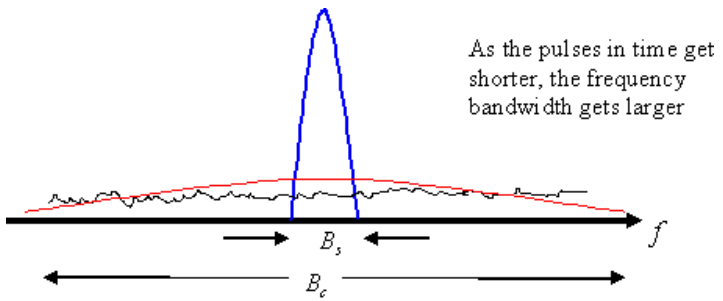




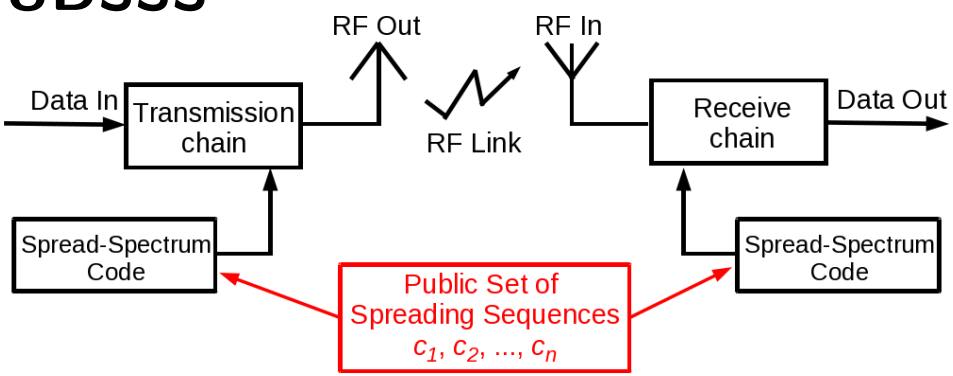
Uncoordinated DSSS (UDSSS)

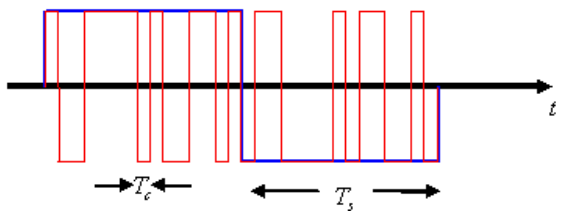
- **DSSS**





- **UDSSS**



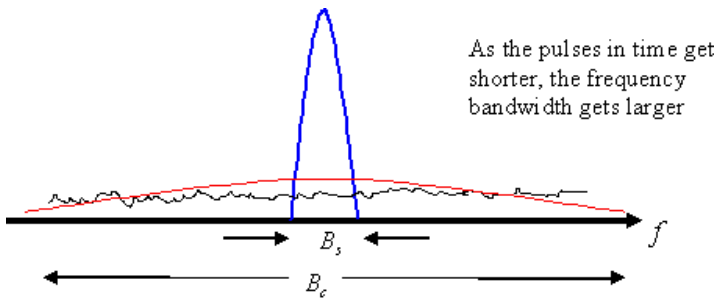


Uncoordinated DSSS (UDSSS)

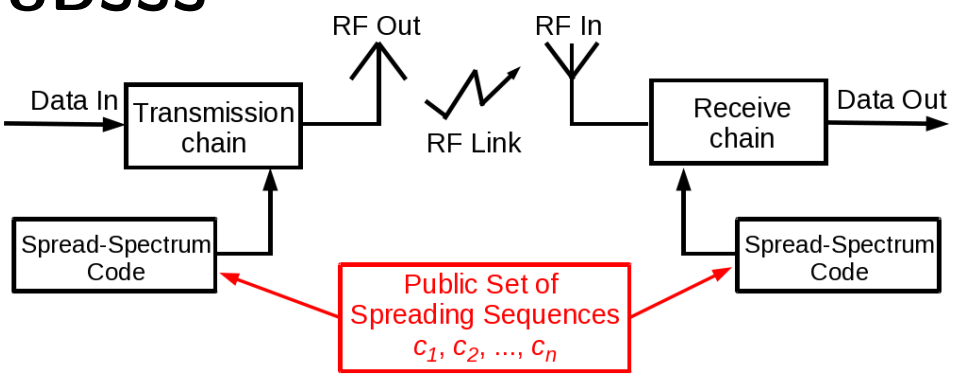
- Public set C of spreading sequences

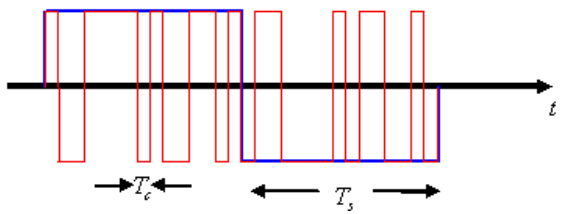
Sender randomly selects sequence $c_s \in C$ to spread message M

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



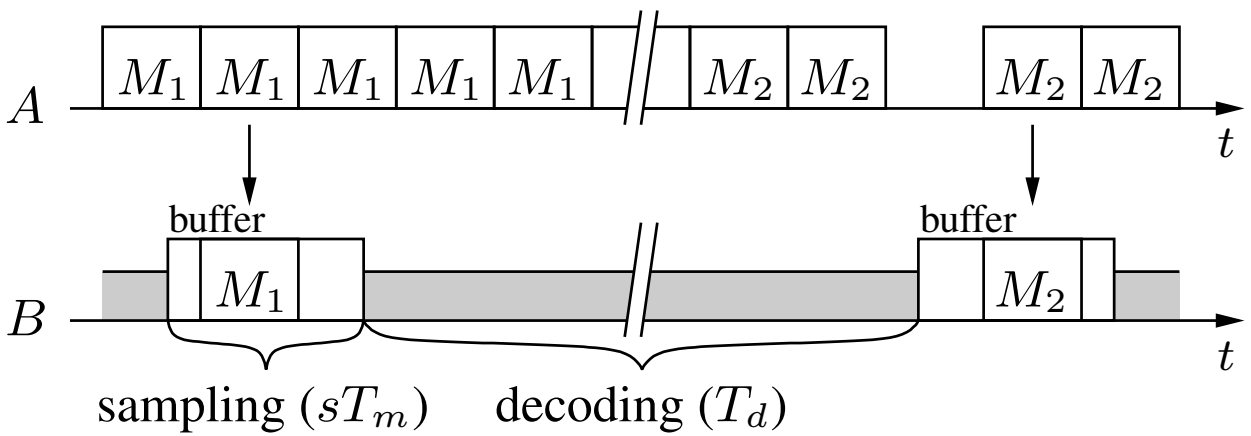
- **UDSSS**





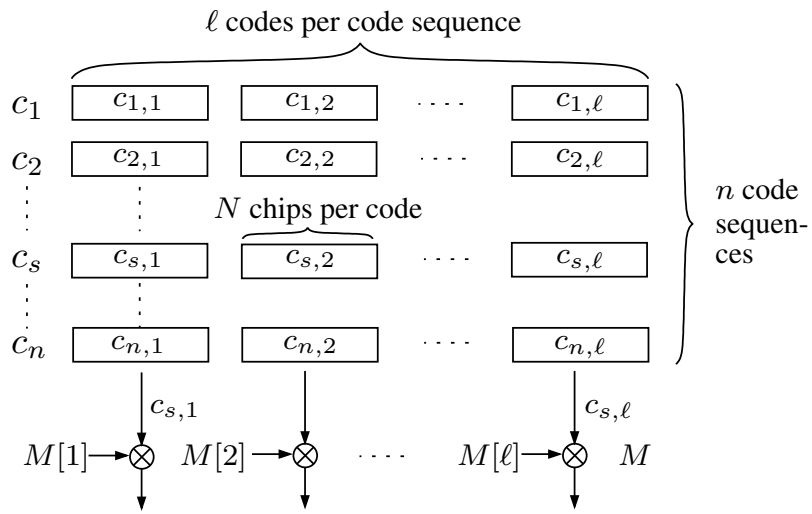
UDSSS Sender Side

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



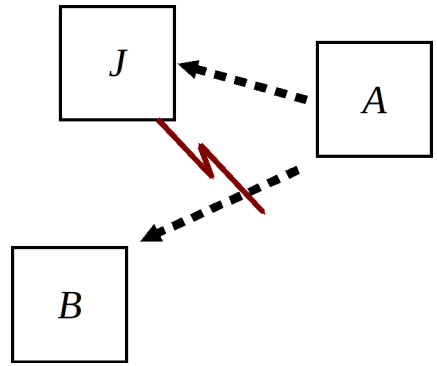
UDSSS Code Set & Despreading

- 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

Attacker Analysis



- **Attacker goal**: To prevent communication
- **Attacker types**
  - Non-reactive jammers blindly jam part of the spectrum
  - Reactive jammers sense for ongoing transmissions
    - Decoding jammers: try to find the used spreading codes and construct the corresponding jamming signal
    - Repeater jammers: intercept the signal and re-radiate it without knowledge of the used spreading codes
- **Attacker strength**: Jamming probability p_j (with respect to a given message transmission)

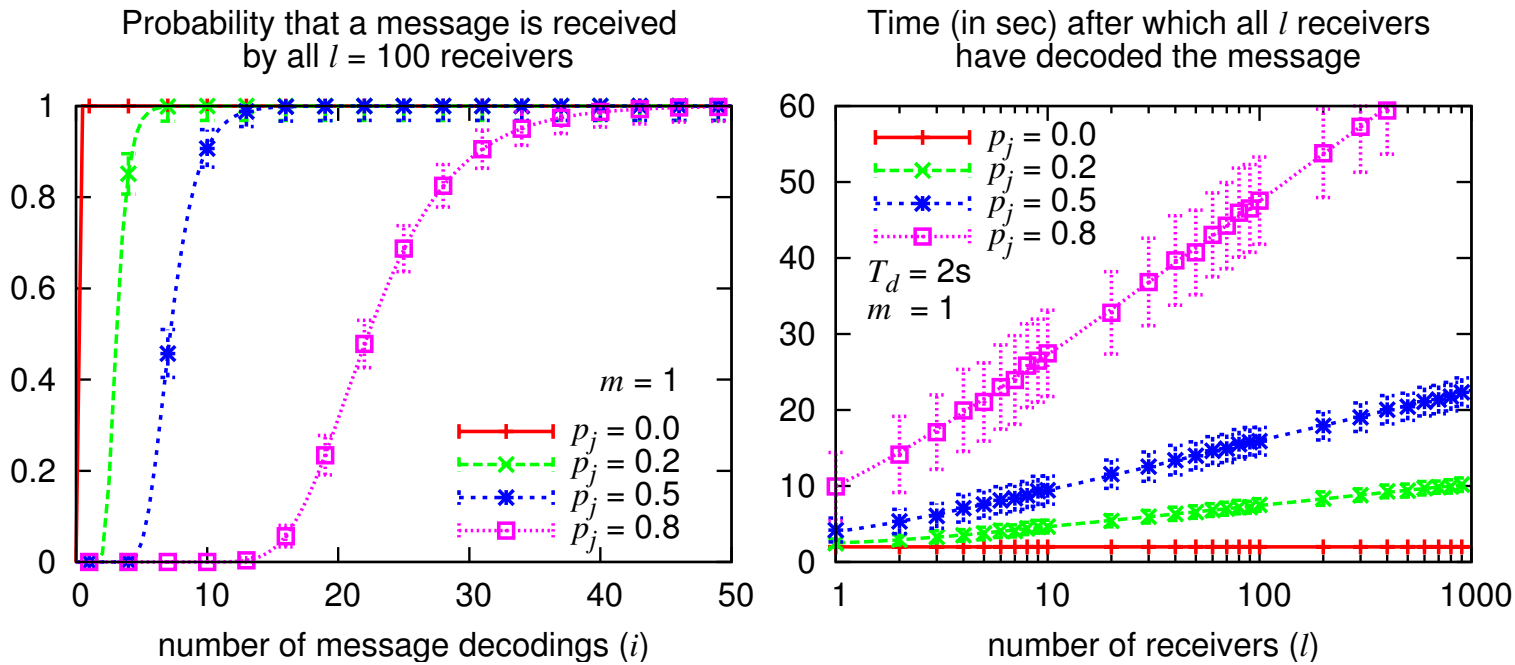
Performance Evaluation

- Evaluation metric: Message transmission time
  - One receiver: Expected time for message recovery at a receiver with jamming ($p_j > 0$) and without jamming ($p_j = 0$)
  - Multiple receivers: Expected time until all l receivers have received the message (for independent receptions) under p_j

- One receiver: $\boxed{T_r \approx T_s + T_d = \frac{2|M|N}{R} + \frac{\frac{n}{2}kqN|M|+|M|}{\Lambda_B(N)}}$
  - $R = 1/T_c$ chip rate
  - q samples per chip
  - $\Lambda_B(N)$: # bit despreading operations that the receiver can perform per second
  - despread k bits before decision on code sequence, etc.

Analytical Evaluation and Simulation

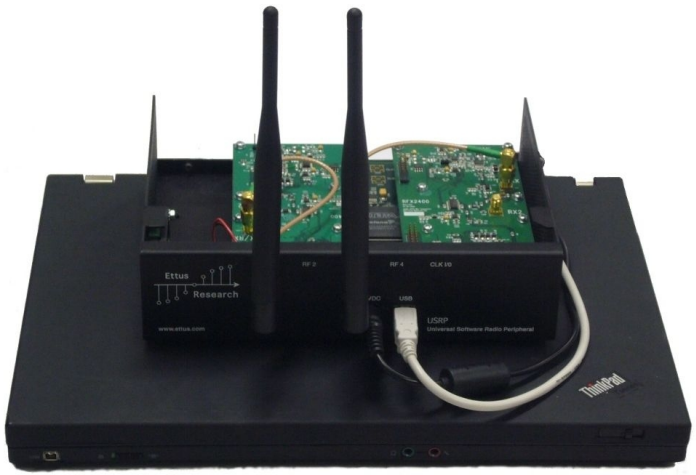
- Multiple (l) receivers

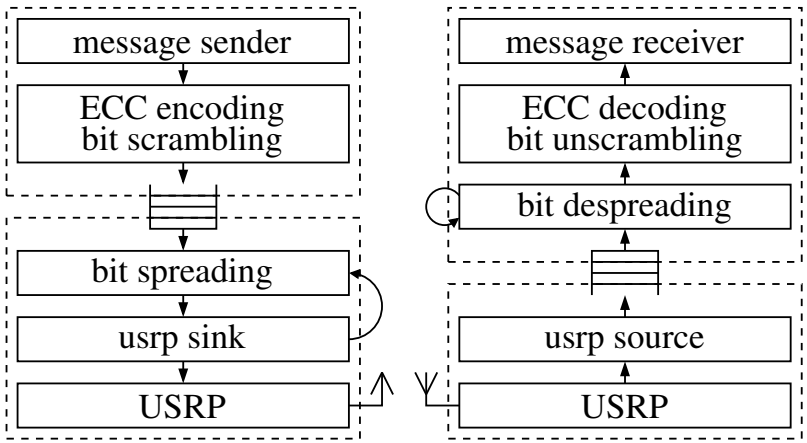


- UDSSS can be enhanced to yield the same performance as (non-synchronized) DSSS in the absence of jamming by two parallel signal transmission using $C_1 = \{c_1\}$ and C_2

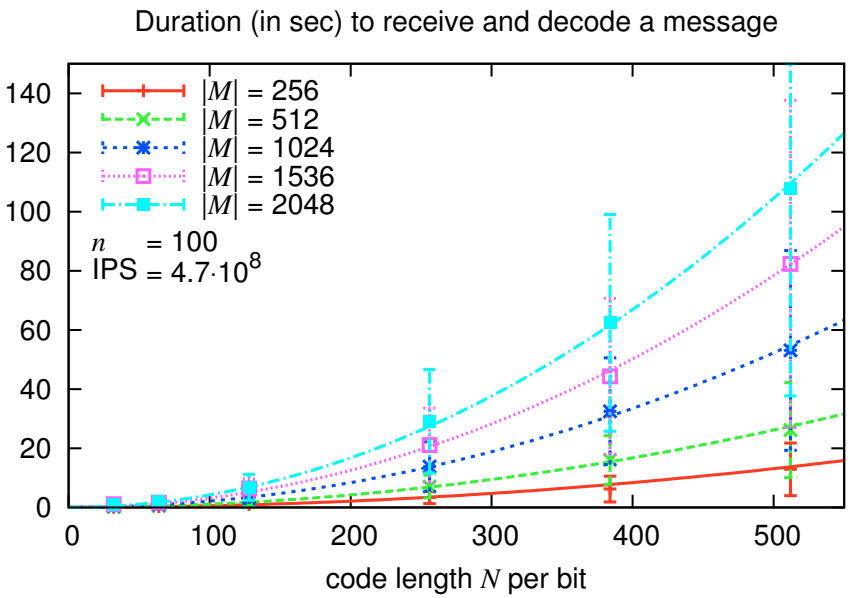
Implementation

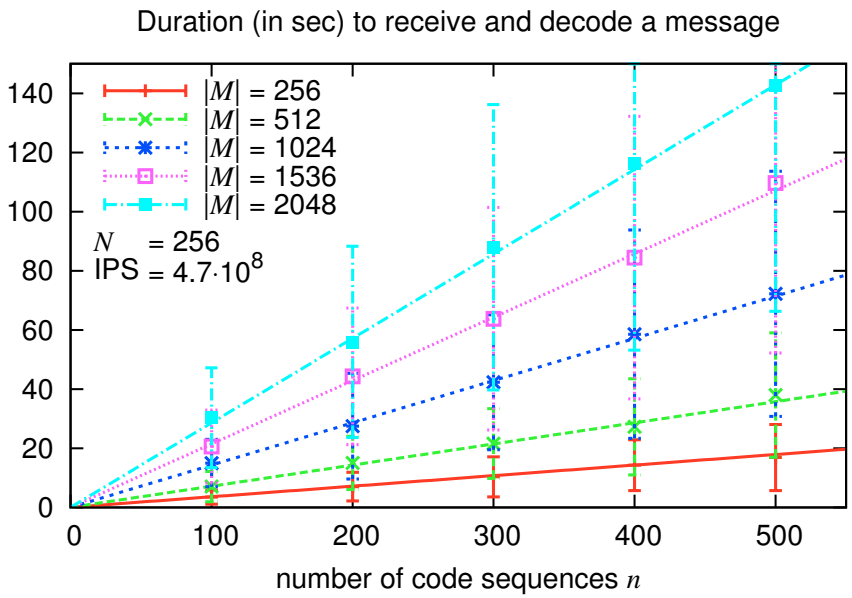
- Prototype implementation of UDSSS on USRP/GnuRadio
  - Carrier frequency of 2.4 GHz
  - (8,4)-Hamming-code ECC
  - 2 USRPs positioned indoors at a distance of around 5 m





Implementation Results

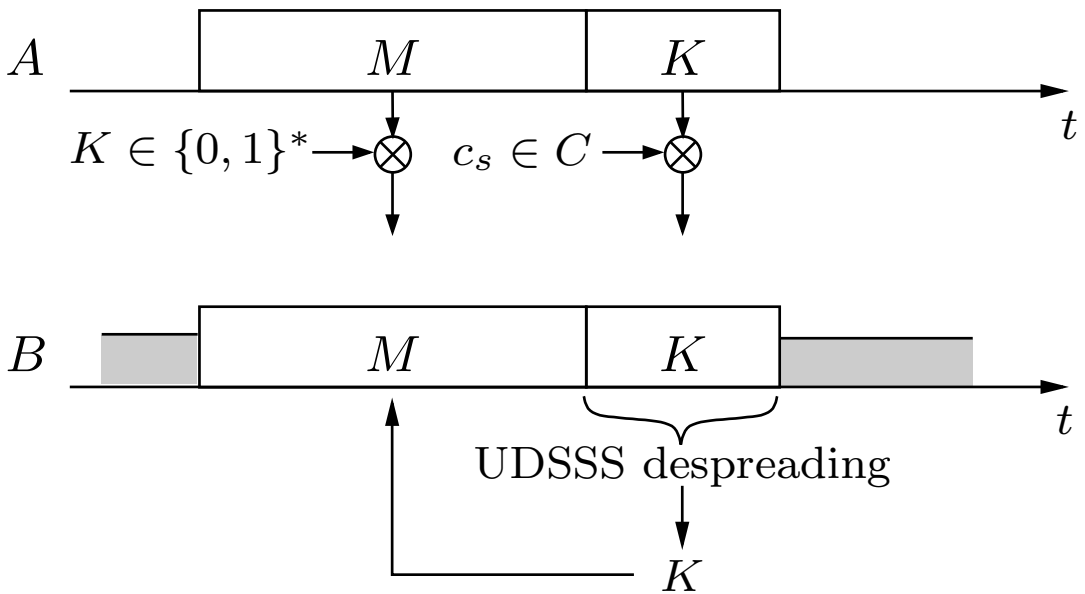




- Increasing the processing gain (i.e., N) is more harmful to the latency/throughput than increasing the code set (i.e., n)

UDSSS Optimization

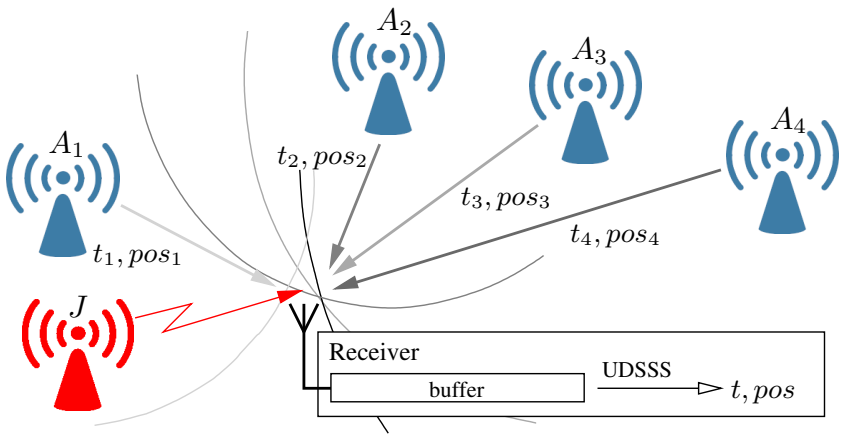
- **Idea**: Use UDSSS to transmit the spreading key only
- **Trick**: First transmit message M using a random spreading code K, then transmit the spreading code K using UDSSS

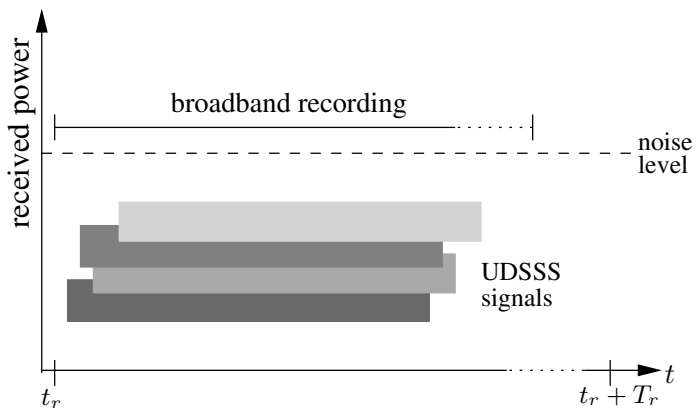


- **Advantages**: Smaller spreading code set. Quicker decoding. Longer messages. More flexible security level.

UDSSS Application: Navigation Signal Broadcasts

- 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

Concluding Remarks

- We tackled the anti-jamming broadcast problem: anti-jamming broadcast communication without pre-shared secrets such that devices cannot jam the reception of other receivers
- Uncoordinated Spread Spectrum techniques are a solution to the anti-jamming broadcast problem
  - UDSSS
  - ZPK-DSSS [Jin et al, MobiHoc09]
  - UFH [Strasser et al., S&P08], [Strasser et al., MobiHoc09], and [Slater et al., WiSec09]
- Basic idea: randomize the spreading operation (random code selection)
- Application: e.g., anti-jamming navigation signal broadcasts

Questions