Concurrent Code Spread Spectrum:

Theory and Performance Analysis of Jam Resistant Communication Without Shared Secrets

William Louis Bahn

Ph.D. Dissertation

Electrical Engineering

Department of Electrical and Computer Engineering

University of Colorado at Colorado Springs

2012

Made possible by….

Committee

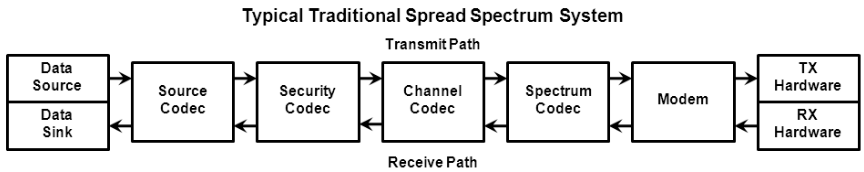
- Mark A. Wickert – Chair
- Rodger E. Ziemer
- Charlie Wang
- C. Edward Chow
- Xiaobo Zhou
- Leemon C. Baird III

ACADEMY CENTER FOR CYBERSPACE RESEARCH

This work was sponsored, in part, by the 688th Information Operations Wing (IOW), Lackland AFB, TX, and was performed at the Academy Center for Cyberspace Research (ACCR) at the United States Air Force Academy.

This material is based on research sponsored by the United States Air Force Academy under agreement number FA7000-10-2-0043. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the United States Air Force Academy or the United States Government.

A typical spread spectrum system uses a spreading code to achieve jam resistance



Source Codec – Reduces information redundancy
Security Codec – Provides confidentiality, integrity, and authentication
Channel Codec – Adds redundancy that permits channel noise to be mitigated
Spectrum Codec – Spread the signal's bandwidth to effectively hide it

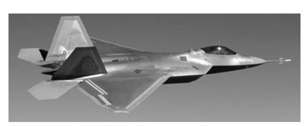
The spreading code defines a channel independent of the message.
An attacker must know the spreading code to find the channel.

We Have a Key Management Problem

Example: STU

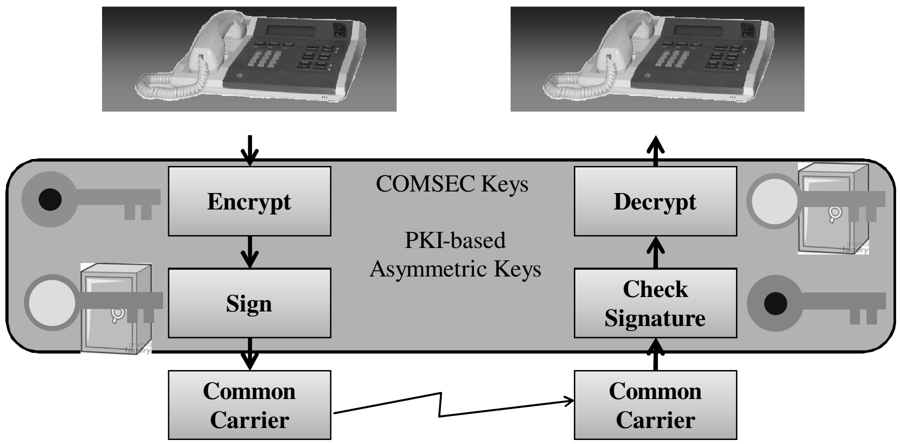
- Occasional keying
- Different key for everyone
- Usually works

Example: Have Quick

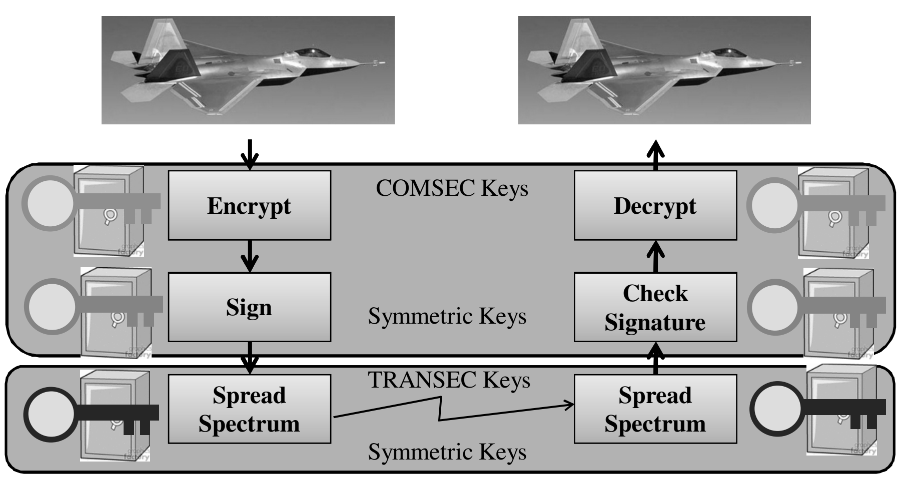


- Daily keying
- Same key for everyone
- Often fails

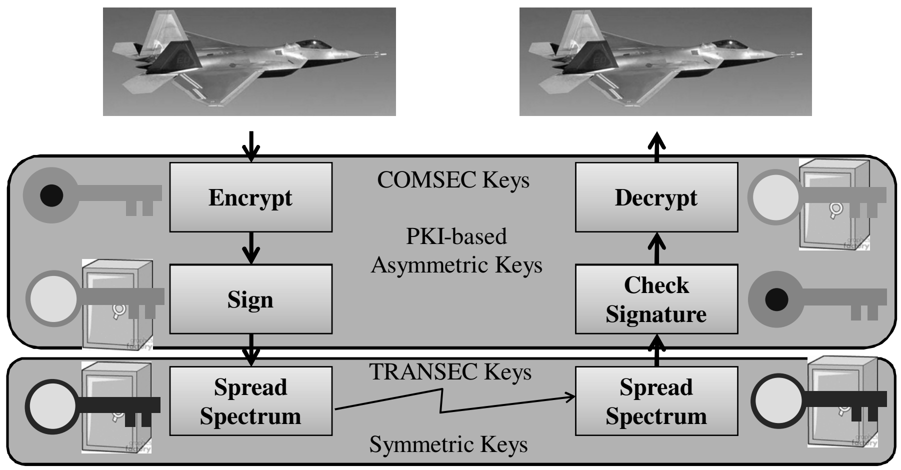
A Public Key Infrastructure (PKI) can provide confidentiality, integrity, and authentication.



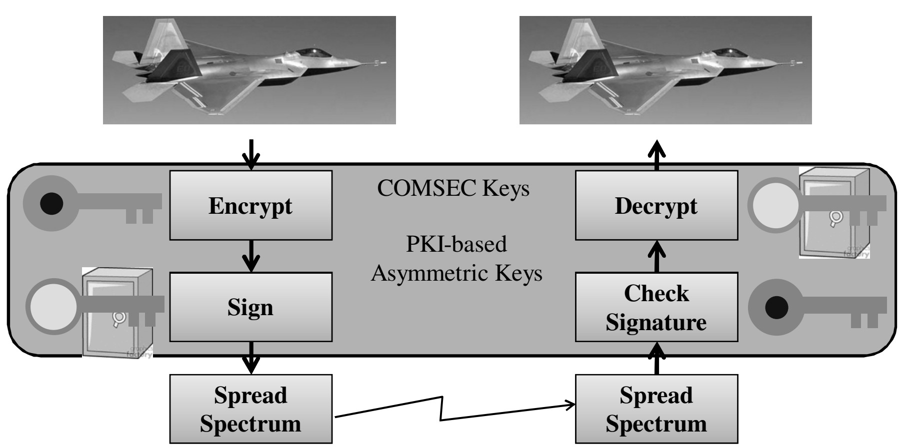
Tactical Comms such as Have Quick



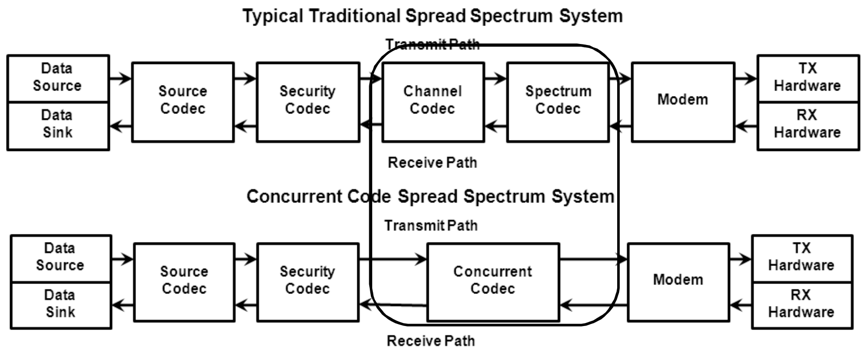
Applying PKI to Tactical Comms



But what if we had KEYLESS Jam Resistance?



A concurrent codec combines channel coding and spectrum spreading



The spreading code defines a channel that depends on the message.
An attacker must know the spreading code and the message to find the channel.
The receiver must merely detect that a message's channel is in use.
Asymmetric channels make suppressing channel activity very difficult.

What challenges face Keyless Jam Resistance?

The optimal jamming signal is another valid signal

- With no key, jammer can transmit valid signals.
- We must assume they can align with legitimate signal.
- Causes extreme difficulty for receivers, which can't decide which valid signal is the legitimate one.
- In traditional systems, two valid signals of similar power make recovery of either impossible.
- Conventional wisdom: Secrets are necessary!
- Unconventional wisdom: Don't be so traditional!

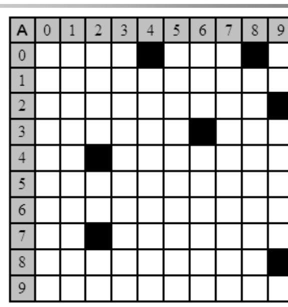
KEYLESS SPREAD SPECTRUM: So how is it possible?

There are two intertwined aspects of Keyless Jam Resistance

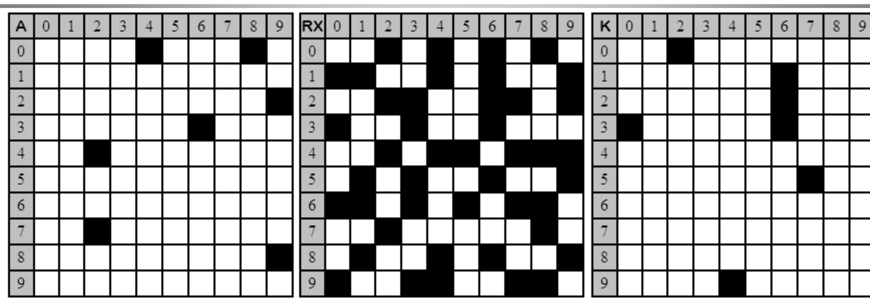
- Coding/Decoding
- Transmitting/Receiving

Let’s build a toy system…

- Codeword for Message A →
- Message space – 32 messages.
- Each codeword is 7 marks.
- 100 possible mark locations.
- Code space – 16 billion codes.
- Message is contained in the signal only if ALL marks associated with that message are present.
- Signal may contain multiple overlaid messages.



Which, if either, is present?



'A' is present

'K' is absent

Are any others present?

Decoding is different than performing a membership test

- Membership test: Is a given message present?
- Signal decode: List all messages that are present.

What if a superimposed code is used?

$$Position_i = \text{Hash}_i(message); \quad 1 < i < n$$

- Message is passed through *n* hash functions, each producing one of the *n* mark locations.
- Each message consists of an independent set of marks.
- With superimposed codes:
  - Membership tests are easy.
  - Decodes are difficult (usually requiring exhaustive search)

Concurrent Codes

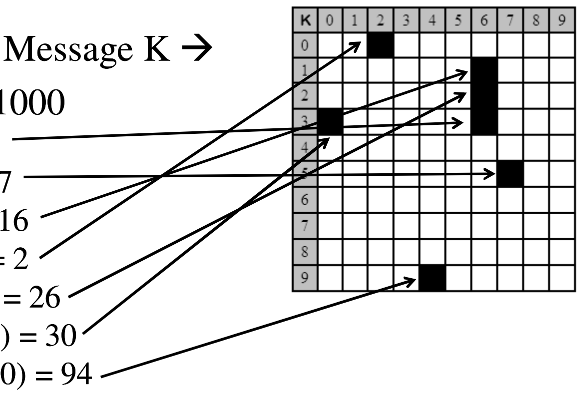
Superimposed codes that can be efficiently decoded.

$$Position_i = \text{Hash}(message[1:i]); \quad 1 < i < n$$

- Each of the *n* possible message prefixes are passed through the same hash function.
- Message codewords are not independent.
- With concurrent codes:
  - Membership tests are easy.
  - Decodes are easy (performed in linear time)

Let's build a codeword

- Codeword for Message K →
- Message: 0101000
  - Hash(0) = 36
  - Hash(01) = 57
  - Hash(010) = 16
  - Hash(0101) = 2
  - Hash(01010) = 26
  - Hash(010100) = 30
  - Hash(0101000) = 94



The complete codebook

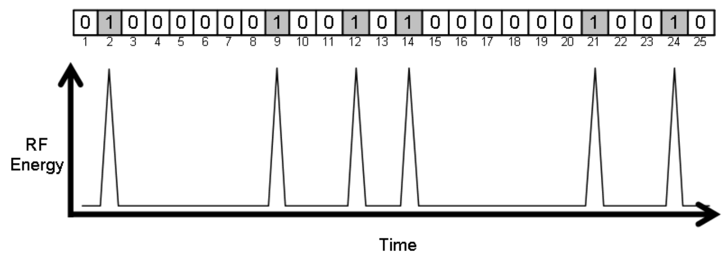
36	89	08	04	42	72	29	A	0000000
				82	46	64	B	0000100
			28	18	48	25	C	0001000
				62	36	88	D	0001100
		91	52	49	01	45	E	0010000
				79	71	38	F	0010100
			13	03	56	12	G	0011000
				98	53	22	H	0011100
	57	16	40	37	47	50	I	0100000
				92	30	76	J	0100100
			02	26	30	94	K	0101000
				78	61	32	L	0101100
		59	22	75	15	80	M	0110000
				85	20	40	N	0110100
			43	31	99	36	O	0111000
				18	67	93	P	0111100
27	19	63	81	14	33	06	Q	1000000
				04	87	41	R	1000100
			46	10	58	66	S	1001000
				69	51	08	T	1001100
		11	07	83	76	28	U	1010000
				54	13	17	V	1010100
			35	09	57	73	W	1011000
				44	39	24	X	1011100
	23	49	11	86	60	05	Y	1100000
				19	53	84	Z	1100100
			72	00	12	46	1	1101000
				67	52	61	2	1101100
		90	24	79	31	99	3	1110000
				44	71	18	4	1110100
			96	25	01	56	5	1111000
				68	88	39	6	1111100

The decode tree for our signal packet: {A,Q,Z} are the messages contained.

36	89	08	04	42	72	29	A	0000000
				~~82~~	46	64	B	0000100
			~~28~~	18	48	25	C	0001000
				62	36	88	D	0001100
		~~91~~	52	49	01	45	E	0010000
				79	71	38	F	0010100
			13	03	56	12	G	0011000
				98	53	22	H	0011100
	~~57~~	16	40	37	47	50	I	0100000
				92	30	76	J	0100100
			02	26	30	94	K	0101000
				78	61	32	L	0101100
		59	22	75	15	80	M	0110000
				85	20	40	N	0110100
			43	31	99	36	O	0111000
				18	67	93	P	0111100
27	19	63	81	14	33	06	Q	1000000
				04	~~87~~	41	R	1000100
			~~46~~	10	58	66	S	1001000
				69	51	08	T	1001100
		11	~~07~~	83	76	28	U	1010000
				54	13	17	V	1010100
			~~35~~	09	57	73	W	1011000
				44	39	24	X	1011100
	23	49	11	86	60	~~05~~	Y	1100000
				19	53	84	Z	1100100
			72	~~00~~	12	46	1	1101000
				67	~~52~~	61	2	1101100
		90	~~24~~	79	31	99	3	1110000
				44	71	18	4	1110100
			~~96~~	25	01	56	5	1111000
				68	88	39	6	1111100

You now know Standard BBC!

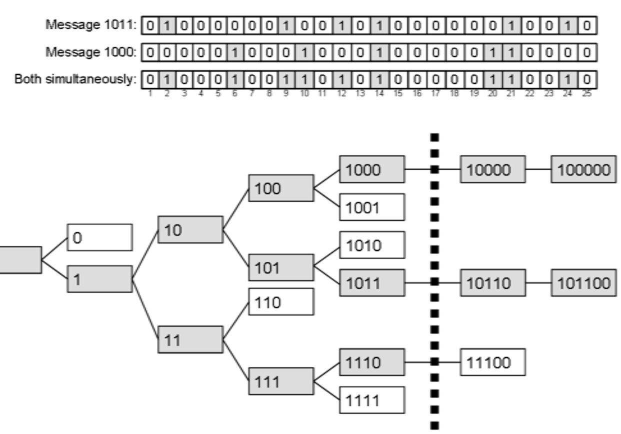
Standard BBC: Encoding is easy



s	H(s)
1	21
10	9
101	24
1011	2
10110	14
101100	12

- Append K zero bits to the end of an L-bit message.
- Run each of the (L+K) prefixes through a hash function.
- Use each hash output to place a mark in the codeword.

Standard BBC: Decoding is fast

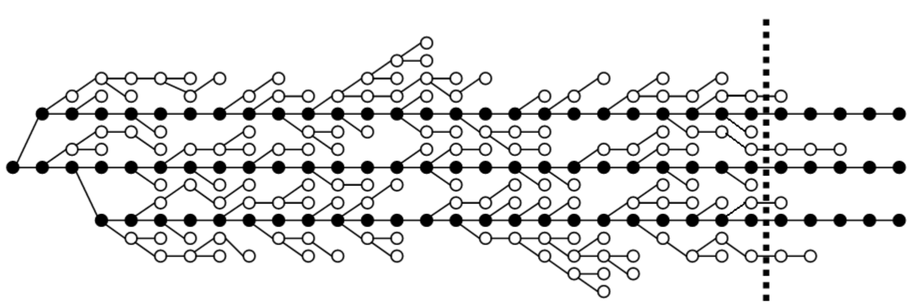


s	H(s)
0	4
1	21
10	9
11	21
100	20
101	24
110	16
111	2
1000	14
1001	1
1010	15
1011	2
1110	14
1111	23
10000	6
10110	14
11100	13
100000	10
101100	12

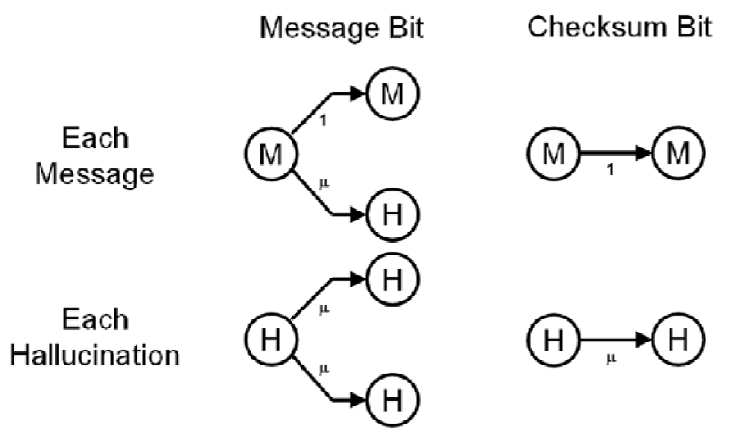
Typical BBC parameters used to date provide good flexibility.

- Thousand bit messages ($L=2^{10}$)
- Million bit codewords ($C=2^{20}$)
- Thirty bit checksum ($K=2^5$)
- 33% mark density threshold

Decode trees typically have many short-lived false branches.



The performance is controlled by the time spent hallucinating.

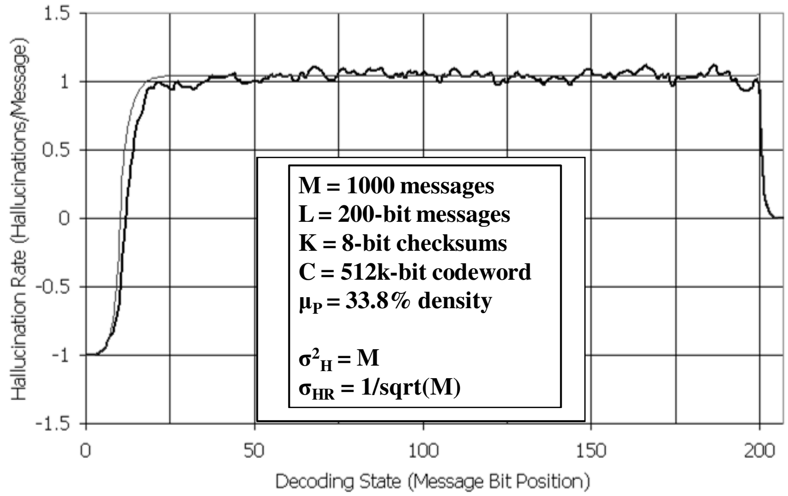


Packet densities <50% are fine

$$H_{SS} = M\left(\frac{\mu}{1-2\mu}\right)$$

- If denominator goes to zero …. That's BAD!
  - Critical Density: $\mu_C = 50\%$
- If $\mu = 33\%$
  - One hallucination per valid message

Performance matches theory



We can enhance BBC three ways

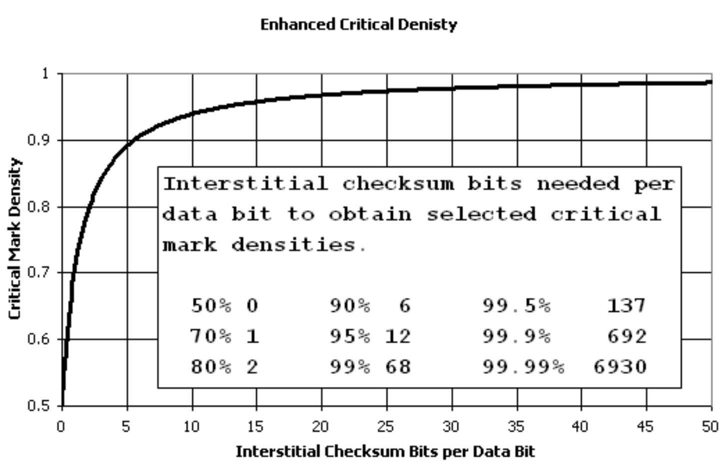
- Interstitial Checksum Bits
  - Arbitrarily high critical density.
  - Smaller hallucinogenic load.
- Multi-mark BBC
  - Can tolerate missed marks
- Multi-bit BBC (M-ary BBC)
  - Can improve coding efficiency

BUT NOTHING COMES FOR FREE!

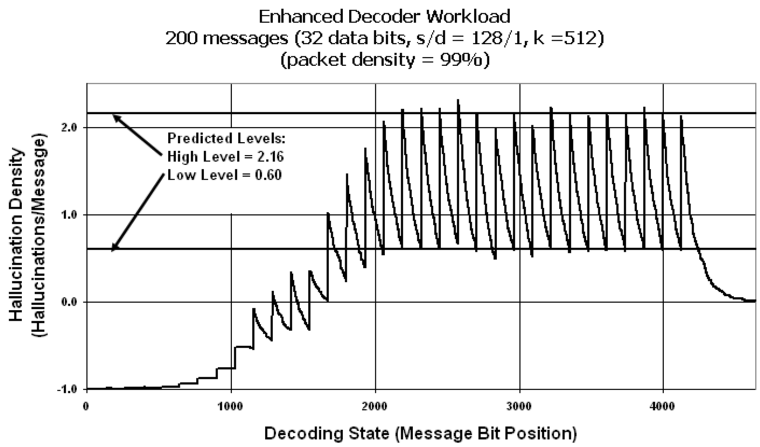
Interstitials are fun, but of limited practicality

- More bits > more marks > higher density
- Codeword density grows faster than critical density
- Lower hallucination load can lead to faster decodes

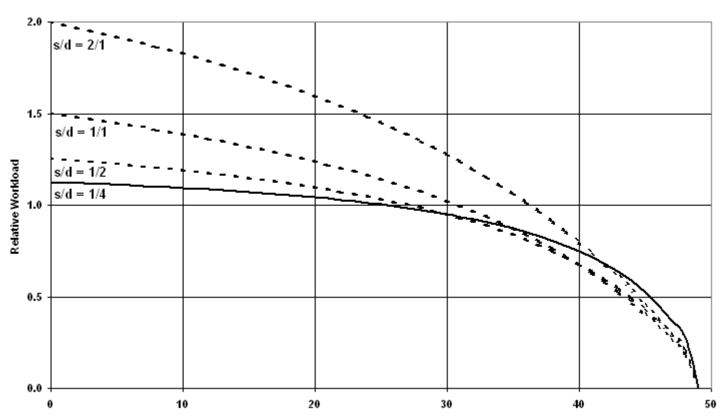
Very rapid improvement for first few interstitial checksum bits



Very high density codewords can still decode efficiently



Mild use of interstitials can decrease absolute decode effort.



Multi-bit BBC is probably not useful

- Nice parallel to M-ary modulation schemes
- Critical density inversely proportional to block size
- Decoding burden exponential with block size
- Quadrature (2-bit/mark) may be useful

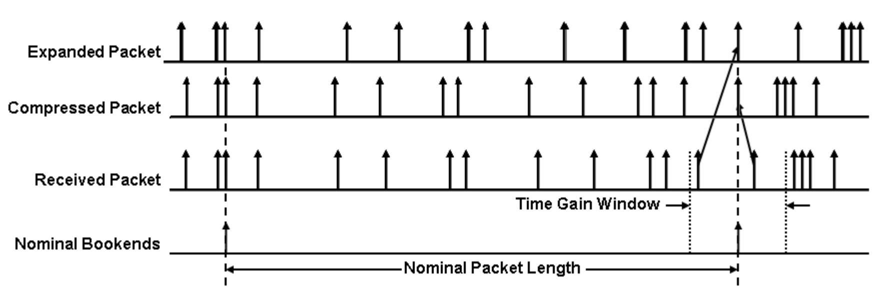
Q-of-Y Multi-mark BBC is probably useful

Q	Y	µ crit	PLR=1%
1	1	50%	10 ppm
1	2	29%	3170 ppm
2	2	71%	5 ppm
1	3	21%	21580 ppm
2	3	50%	1831 ppm
3	3	79%	3 ppm

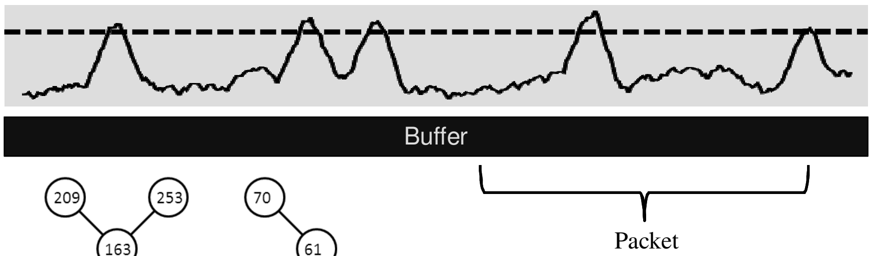
What about practical considerations?

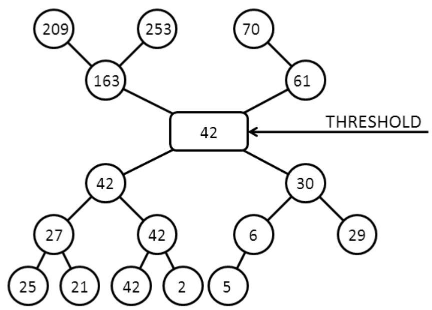
- Packet identification
  - Place "Bookend Marks" in each packet
  - Each received mark is treated as a potential packet start
- Symbol timing
  - Leverage intrinsic tolerance for space errors
  - Leverage packet-level decoding to compensate
- Threshold level
  - Running statistic threshold forces optimal threshold
- Hash function performance
  - Use task-specific hash function (Glowworm)
- Multipath
  - Intrinsic tolerance – echoes are just additional messages

Bookend marks permit compensation for significant oscillator mismatch



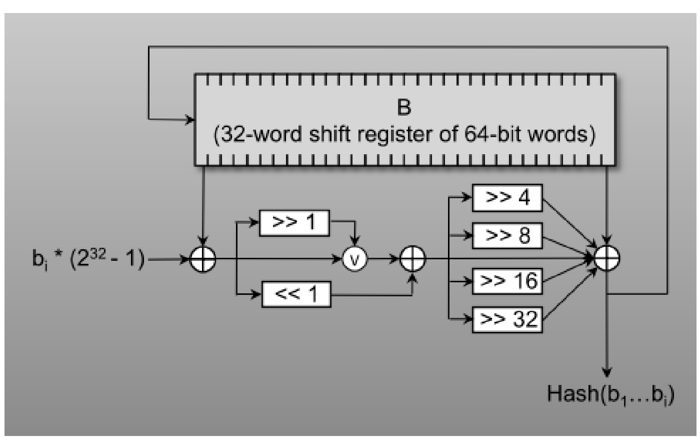
Running statistic threshold can set the best threshold for EACH packet.





Log(n) time per packet!

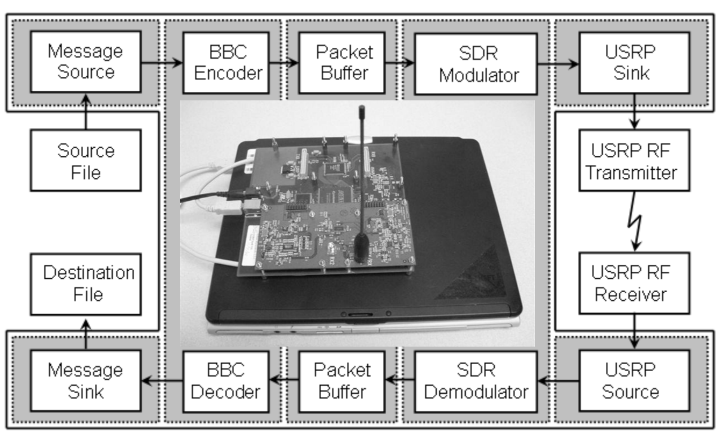
The Glowworm hash is structurally simple and efficient in both hardware and software



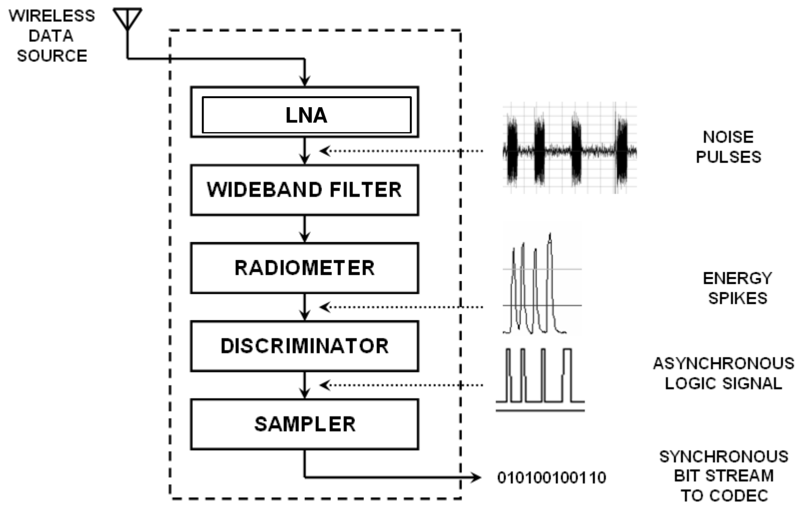
40

Does it really work?

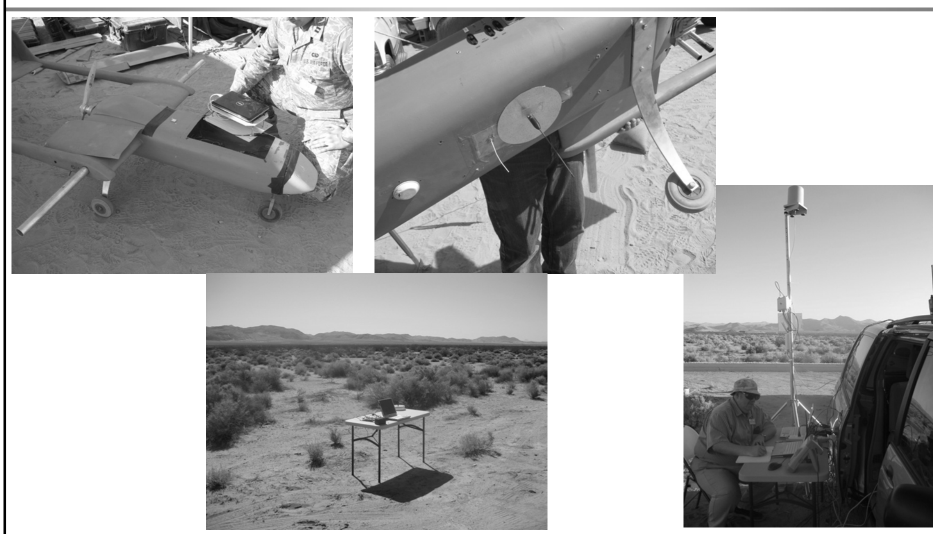
RF demonstrator based on Ettus USRP software-defined radio



Simple receiver leaves little for attacker to attack

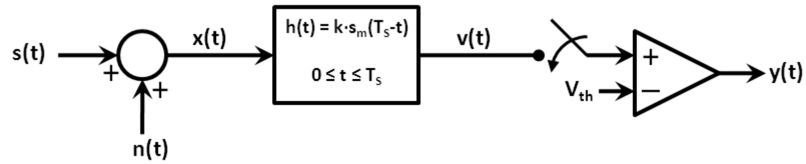


NAWS China Lake – Fun in the Sun!

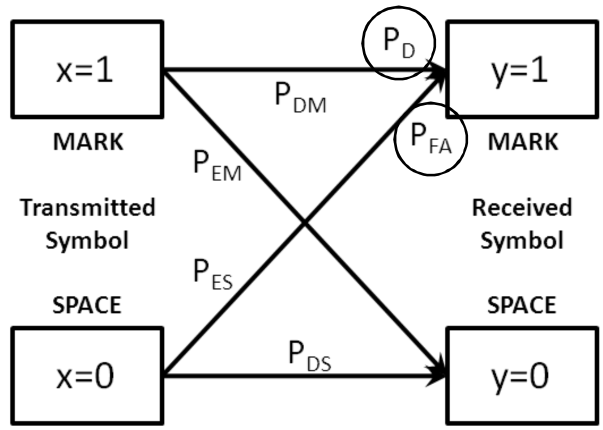


Is it really Jam Resistant?

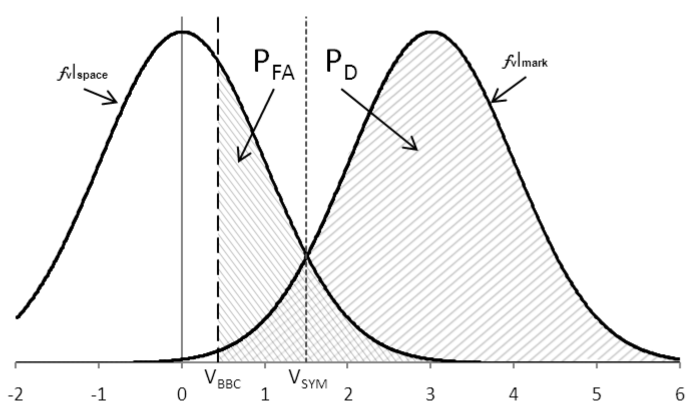
The analyzed receiver used a typical matched-filter detector



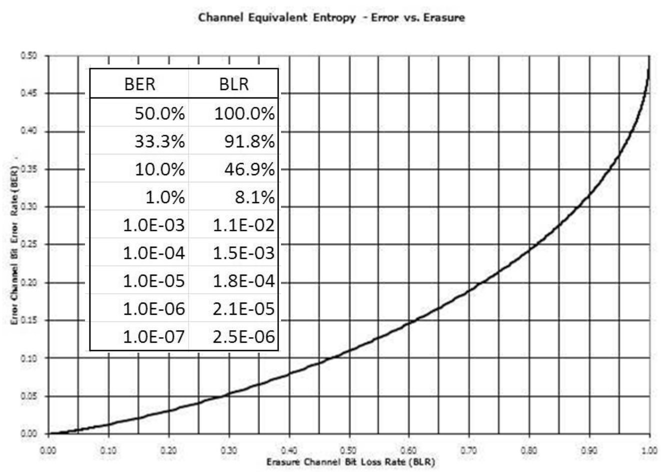
The same translation paths exist, but radar terminology convenient



The BIG difference: Threshold set well into the noise



Concurrent codes create erasure channels, not error channels

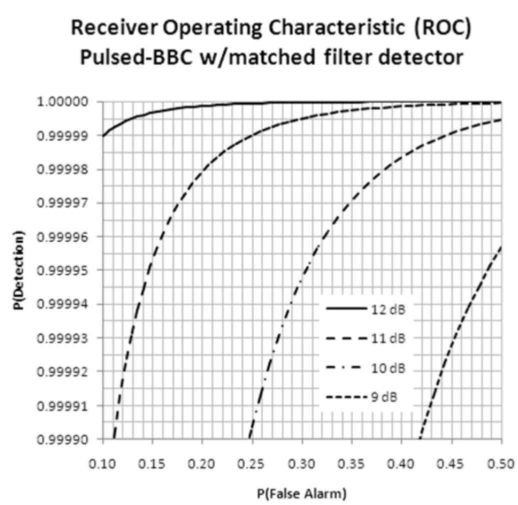


BER	BLR
50.0%	100.0%
33.3%	91.8%
10.0%	46.9%
1.0%	8.1%
1.0E-03	1.1E-02
1.0E-04	1.5E-03
1.0E-05	1.8E-04
1.0E-06	2.1E-05
1.0E-07	2.5E-06

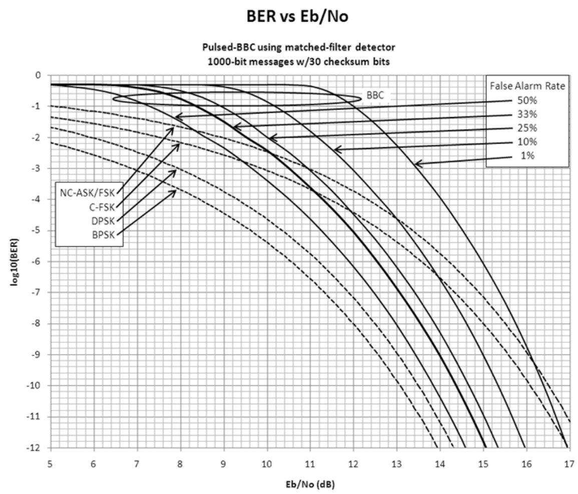
BER of 1% achieved with P_{FA} = 33% & SNR/bit = 9.6 dB

$P_D = (1\text{-PLR})^{1/N}$

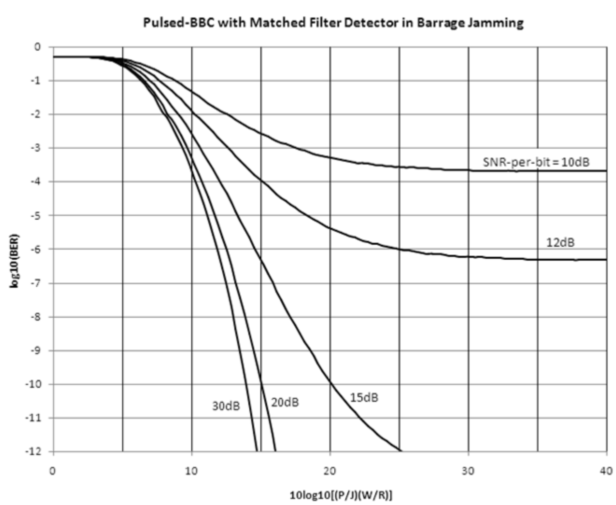
N=1032-mark packets
- **BER = 1%**
  - **PLR = 8.1%**
  - **P_D = 0.99992**
- **P_{FA}=33%**
  - **SNR/bit = 9.6 dB**



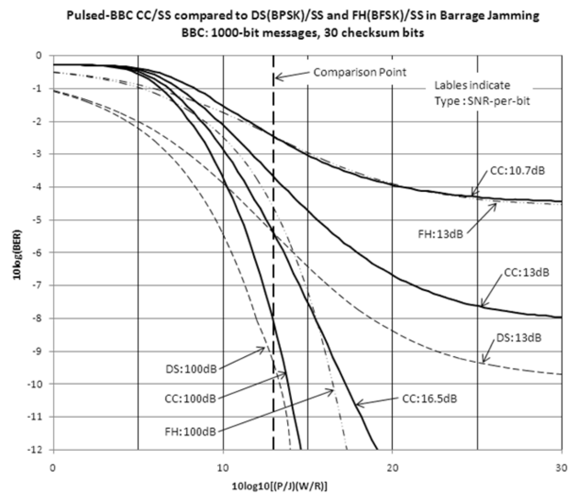
CCSS performs between FSK and PSK (SNR/bit > 11dB)



SNR/bit >~20dB realizes near-optimal performance



CCSS performs better than FHSS but not as well as DSSS



Where can we go from here?

- RF performance analysis
  - Mark waveforms (e.g., LFM chirps, Golay sequence)
  - Detector options (e.g., radiometer, Golay correlator)
  - Asynchronous issues (e.g., symbol alignment)
  - Optimal waveform jamming (e.g., mark erasure, packet construction)
- Comparison with other forms as an unreliable erasure channel
- RF implementations of the various waveform/detector combos
- FPGA/ASIC implementations of radios and/or building blocks
- MAC-less protocol development
- Other application areas (e.g., RFID, SINCGARS Fill Device)
- Other questions (e.g., performance guarantees, non-BBC codes)

What was done and what I did

- Concurrent codes are a notable extension to superimposed codes
  - Efficient decoding is something that has not existed
  - Potentially opens door to many previously ill-suited applications
- Concurrent code spread spectrum offers new capabilities
  - Comparable jam resistance without shared secrets
  - Potentially simpler MAC-layer protocols (or even MAC-less protocols)
- My contributions
  - I was the primary contributor for:
    - RF hardware , software, or analysis
    - Interstitial checksum bits and multi-mark
    - Oscillator mismatch and jitter compensation
  - I was heavily involved in the collaboration on most other aspects
  - I contributed least to the “hard core” theoretical/mathematical aspects

Peer Reviewed Publications (1/2)

1. L. C. Baird, III, M. C. Carlisle, and W. L. Bahn, “Unkeyed jam resistance 300 times faster: The Inchworm hash,” in Proc. 2010 IEEE Military Communications Conference (MILCOM10), Nov. 2010, p. CD.
2. L. C. Baird, III, D. L. Schweitzer, W. L. Bahn, and S. Sambasivam, “A novel "Visual Cryptography" coding system for jam resistant communications,” Journal of Issues in Informing Science and Information Technology, vol. 7, pp. 495--507, 2010.
3. W. L. Bahn, L. C. Baird, III, and M. D. Collins, “Oscillator mismatch and jitter compensation in concurrent codecs,” in Proc. 2008 IEEE Military Communications Conference (MILCOM08), Nov. 2008, p. CD.
4. W. L. Bahn, L. C. Baird, III, and M. D. Collins, “Jam-resistant communications without shared secrets,” in Proc. 3rd International Conference on Information Warfare and Security (ICIW08), Apr. 2008, p. CD.

Peer Reviewed Publications (2/2)

5. W. L. Bahn, L. C. Baird, III, and M. D. Collins, “The use of concurrent codes in computer and digital signal processing education,” Journal of Computing Sciences in Colleges, vol. 23, no. 1, pp. 174{180, Oct. 2007.
6. D. L. Schweitzer, L. C. Baird, III, and W. L. Bahn, “Visually understanding jam resistant communication,” in Proc. 3rd Intl. Workshop on Visualization for Computer Security (VizSec), Oct. 2007.
7. W. L. Bahn, L. C. Baird, III, and M. D. Collins, “Impediments to systems thinking: Communities separated by a common language,” in Proc. 4th Intl. Conf. on Cybernetics, Information Technologies, Systems and Applications (CITSA), vol. III, Jul. 2007, pp. 122--127.
8. L. C. Baird, III, W. L. Bahn, M. D. Collins, M. C. Carlisle, and S. C. Butler, “Keyless jam resistance,” in Proc. 8th Annual IEEE SMC Information Assurance Workshop (IAW), Jun. 2007, pp. 143—150.

USAFA Technical Reports (1/2)

1. W. L. Bahn, “A field demonstration of unkeyed jam resistance,” United States Air Force Academy, Academy Center for Cyberspace Research, Tech. Rep. USAFA-TR-2010-ACCR-04, Dec. 2010.
2. L. C. Baird, III and W. L. Bahn, “Parallel BBC decoding with little interprocess communication,” United States Air Force Academy, Academy Center for Cyberspace Research, Tech. Rep. USAFA-TR-2009-ACCR-01, Nov. 2009.
3. L. C. Baird, III and W. L. Bahn, “An O(log n) running median or running statistic method, for use with BBC jam resistance,” United States Air Force Academy, Academy Center for Cyberspace Research, Tech. Rep. USAFA-TR-2008-ACCR-03, Nov. 2009.
4. L. C. Baird, III and W. L. Bahn, “An efficient correlator for implementations of BBC jam resistance,” United States Air Force Academy, Academy Center for Cyberspace Research, Tech. Rep. USAFA-TR-2008-ACCR-02, Nov. 2009.

USAFA Technical Reports (2/2)

5. W. L. Bahn and L. C. Baird, III, "Hardware-centric implementation considerations for BBC-based concurrent codecs," United States Air Force Academy, Academy Center for Cyberspace Research, Tech. Rep. USAFA-TR-2008-ACCR-03, Dec. 2008.
6. W. L. Bahn and L. C. Baird, III, "Extending critical mark densities in concurrent codecs through the use of interstitial checksum bits," United States Air Force Academy, Academy Center for Cyberspace Research, Tech. Rep. USAFA-TR-2008-ACCR-02, Dec. 2008.
7. L. C. Baird, III and W. L. Bahn, "Security analysis of BBC coding," United States Air Force Academy, Academy Center for Cyberspace Research, Tech. Rep. USAFA-TR-2008-ACCR-01, Dec. 2008.
8. L. C. Baird, III, W. L. Bahn, and M. D. Collins, "Jam-resistant communication without shared secrets through the use of concurrent codes," United States Air Force Academy, Academy Center for Information Security, Tech. Rep. USAFA-TR-2007-01, 2007

QUESTIONS?