Jam Resistant Communications without Shared Secrets

William L. Bahn
Leemon C. Baird III
Michael D. Collins
United States Air Force Academy, Colorado, USA
William.Bahn@usafa.edu
Leemon.Baird@usafa.edu
Mike.Collins@usafa.edu

Abstract: Information security is only attained when all four of the classic information assurance goals - confidentiality, integrity, authenticity, and availability – are attained. The work presented focuses exclusively on the availability goal; in particular, maintaining the availability of the physical layer link in an omnidirectional wireless communications system. Hostile jamming is a direct attack on the availability of an information resource. As the United States develops the Global Information Grid (GIG), key management will be a challenging problem. While much progress has been made utilizing asymmetric cryptography and a Public Key Infrastructure (PKI) to manage keys that encrypt, decrypt, and authenticate data, the keys used to protect the physical communication layer from hostile jamming are highly vulnerable. In particular, battlefield ad hoc wireless networks are at significant risk. Preferred jam resistant methods rely on highly-directional links (e.g., beam-forming antennas, lasers, or physical cable), but realistic operational environments will continue to rely heavily on omnidirectional radio frequency (RF) links as well. Presently, such links rely on spread spectrum techniques for jam-resistance, all which currently require symmetric (shared-secret) keys. While shared-secret schemes are workable in small networks, the scale and nature of theater-wide, mobile, ad-hoc wireless networks will quickly overwhelm any practical key management strategy. Not only will the initial distribution of keys be difficult, but preventing keys from being compromised and re-keying when the inevitable compromises occur will place extreme burdens on the system. In addition, commercial systems such as cellular phone networks face similar challenges as their need for jam resistance becomes more evident. Furthermore, public-access systems such as the civilian side of the Global Positioning Satellite (GPS) system preclude reliance on secret keys since, by definition, the pool of authorized users includes everyone. Yet while civilian GPS is recognized as having little jam-resistance, the present trend calls for increased reliance on it for civil aviation. The availability goal in such large-scale and/or public access omnidirectional systems can only be achieved if the physical layer link can remain available in the face of significant hostile jamming even without a shared secret in place. This paper describes the first algorithms that make unkeyed jam-resistant omnidirectional physical links possible: the BBC encoding and decoding algorithms. We describe these algorithms and how they can be implemented in suitable physical layers. Such physical layers, while not secure in and of themselves, can serve as the foundation upon which well-established asymmetric techniques can be used to construct secure communications channels.

Keywords: spread spectrum, jam-resistance, physical layer.

1. Introduction - the growing need for omnidirectional jam resistance

Reliable communication is critical to modern economies and warfare (Foster 2000, Gowens 2000). Many examples exist in which even short term interruptions can prove catastrophic: an airliner low on fuel on final approach in bad weather is at serious risk if navigation signals are interfered with; similarly, a battlefield engagement can go horribly awry if the communications used to distinguish friend from foe are compromised. These are just two examples where malicious jamming at a critical time can result in the loss of life.

The 1996 Federal Radionavigation Plan (FAA 1996) called for the termination, by 2010, of all non-GPS-derived navigation services, with the possible exceptions of NDB (non-directional beacons) in Alaska and military TACAN (Tactical Air Navigation). This plan met with vocal opposition (e.g., ASW 1999), particularly in the aftermath of actual interruptions of GPS navigation signals such as that caused by a malfunctioning Air Force test station that allowed a 5W transmitter to seriously degrade service over a wide area for two weeks (Brewin 1998). The growing awareness of the vulnerability GPS (e.g., Carol 2001) led to a major reversal in the FAA's plans (Sutton 1998) and the 2005 Federal Radionavigation Plan (FAA 2005) reveals significant reflection on the wisdom of using GPS as a sole-means system. However, after specifically acknowledging the susceptibility of GPS to both intentional and unintentional jamming, augmented-GPS systems are already approved as primary-means aerial

navigation systems and the plan still calls for significant reductions and the eventual phase-out of the installed base of other systems.

Turning to the battlefield example, the United States is pursuing the doctrine of net-centric operations with the goal of realizing the Global Information Grid (GIG) (NSA 2008, GAO 2004) that will encompass users at all levels including top-level commanders, soldiers in the field, emplaced sensor networks, and conceivably even individual pieces of equipment such as rifles and artillery shells. While resulting "force multiplier" will allow missions to be carried out by fewer people with less equipment and fewer supplies (Alberts 2000), it also creates a critical reliance on the GIG where any significant disruption can threaten mission accomplishment (Alberts 2002).

Any link that forms a critical path in a mission-critical communication system must be designed and employed with jam-resistance in mind. This is particularly true in adversarial environments where malicious players exist whose objectives would be furthered by the disruption of those links. Furthermore, as reliance on these links increases so too will the motivation to disrupt them.

The remainder of the discussion is meant to apply specifically to omnidirectional RF communication links unless specifically stated otherwise. Having said that, the meaning of "omnidirectional", as it applies to jamming, needs some clarification. Technically, the term omnidirectional refers to transmitters and receivers that have no preferred directional sensitivity. The focus here is on links that, from a jamming standpoint, are "effectively omnidirectional", meaning transmitters and receivers that are equally accessible to friend and foe. Thus, while transmitters and receivers on most satellites are extremely directional in the technical sense, the satellite side of a satellite/terrestrial link is effectively omnidirectional. While the terrestrial side of the link can be highly directional even from a jamming perspective, in at least some cases, such as most commercial GPS receivers, this is not the case.

While it is probably true that most of the GIG will be implemented using highly directional links (few things are more directional than a length of wire or fiber optic cable), the "edge of the GIG" will, in many instances, not only be wireless, but will be highly dynamic Mobile Ad Hoc Networks (MANETs) (Van der Merwe 2007) that will be heavily reliant on omnidirectional RF links.

2. Jam resistance and the scalability problem

Jamming is an attack on the availability of an information resource. Unfortunately, a critical aspect of jam resistance tends to fall through the gaps between the information assurance and the (electrical engineering) communications communities (Bahn 2007). The information assurance community tends to limit its focus on availability issues to denial of service attacks occurring in functional networks (i.e., networks successfully moving bits around); they leave the issue of ensuring physical layer functionality to the communications community. Conversely, the communications community tends to focus on designing physical layers capable of successfully moving bits around under hostile jamming. With both communities intently focused on their portion of the system, physical layer key management tends to get overlooked because it is not technical, but rather operational in nature.

The communications community has developed fairly jam-resistant communication links based on spread spectrum. The information assurance community uses these links upon which to build jam resistant networks. However, these links are only jam resistant if the sender and receiver share information, such as the spreading code, that is kept secret from any potential adversaries (Peterson 1995). This shared information thus constitutes, in the vernacular of the information assurance community, a symmetric key. The communications community generally assumes that the users of the system will install the necessary keys in the radios prior to setting up the communications link. Unfortunately, people in the information assurance community are frequently unaware that the physical layer even has a requirement for symmetric keys precisely because they keys are typically installed in the radios prior to setting up the communications link. The end result is that neither community, at the design level, is widely aware that a daunting key management problem exists for the end user.

The problem is daunting because managing symmetric keys does not scale well (Forouzan 2008). If a network is reliant on only a small number of keys each shared by large fractions of the authorized users, then each key compromise can have a disastrous impact and the probability that a key will be

compromised in any given time period increases rapidly with the number of authorized users. If, instead, the network is built around many keys each shared by only a small number of authorized users then while the compromise of any given key has a limited impact the total number of keys required, and hence the overall rate at which compromises occur, increases with the square of the number of authorized users. Furthermore, either approach requires that keys be distributed via a secure channel. When communication systems involve, say, a few dozens of users these problems are reasonably manageable. However, the GIG potentially involves hundreds of thousands of authorized users, millions of pieces of equipment, and perhaps tens of millions of distributed sensor nodes. At this scale, any conceivable symmetric key management scheme will grind to a halt.

3. A new type of jam resistance and the Informed Jammer threat model

If large-scale jam resistance based on symmetric keys is impractical, then a type of jam resistance that does not rely on symmetric keys must be found. The basic goal can be stated as follows: Transfer a message from sender to receiver in the face of significant hostile jamming even when the jammer possesses all knowledge common to both sender and receiver. Such a jammer is defined, by the authors, to be an "informed jammer" as opposed to the widely defined "smart jammer" who is assumed to be ignorant of any such shared secrets (Peterson 1995). To be precise, an informed jammer is a smart jammer that has gained access to those secrets.

Since no system is jam proof a jam resistant system must be measured against a relevant threat model that not only grants adversaries certain abilities, as described above, but also subjects them to certain limitations. Informed jammers have two limitations: (1) their attacks consist only of radiated RF energy, and (2) they have an average power limit (i.e., total energy over some time period) they cannot exceed. The first is simply to eliminate from the discussion such things as physical attacks, which are part of other threat models. The second is driven by the fact that jammers do not operate in a threat-free environment. In addition to physical limitations on total energy that might be imposed, for instance, on remotely deployed battery-operated jammers, they must take care not to let their transmissions rise above a certain level lest they risk expose to active countermeasures. In the case of civilian jamming this might entail being arrested or fined. In tactical military environments the penalties tend to be more direct and permanent.

One subtle but profound consequence of facing an informed jammer is that, due to the lack of a symmetric key in the physical layer, error correcting codes cannot be used to improve jam resistance. The reason is that error correcting codes strive to take a received vector (assumed to be a corrupted codeword) and identify which codeword is "nearest" to it. For this to work, the received vector must lie within the error correcting distance of the correct codeword. But since the sender and receiver have no shared secrets, the jammer can generate and transmit valid codewords. If we assume that the jammer's choice of codewords is independent of those used by the legitimate sender then, on average, the two codewords will be separated by exponentially many other codewords. Regardless of how the communications channel combines the two codewords to form the received vector, symmetry argues that the received vector is as likely to lie closer to the jammer's codeword as it is to the legitimate sender's codeword and, on average, can be expected to lie roughly equidistant between the two, which still places an exponentially large number of codewords between the received vector and the legitimate one. Thus the likelihood of an error correcting code successfully recovering the legitimate sender's codeword is miniscule.

What is required is a new coding theory that permits the efficient extraction of all possible valid codewords that are consistent with a received packet constructed by the superposition of multiple valid codewords plus channel noise. Aside from being physically realizable, the channel's behavior must be such that the superposition of codewords results in a packet of bits that recognizably contains all of the codewords that were combined and, within its performance envelope, only a relatively small number of others. In most transmission schemes in common use, the bit error probabilities (BER) are symmetric and, as a result, this is not possible. But if the BER is highly asymmetric (as it can be, for instance, with On-Off Keying (OOK) modulation), then it becomes possible to construct a "multiple-access OR-channel" wherein codewords are combined by performing a bitwise OR'ing of the individual codewords to create a packet. A codeword is then considered to be contained in a received packet if and only if all of its HI bits are contained in the packet.

4. The multiple-access OR-channel and the indelible mark

Multiple-access OR-channels are, to a good approximation, physically realizable. Consider, as an example, impulse-based ultra wideband radio used as follows: The sender transmits a high power, short duration pulse of noise at times corresponding to marks (i.e., HI bits) in the codeword and remains silent at times corresponding to spaces (i.e., LO bits). Traditionally, the threshold would be raised in an effort to minimize the overall bit error rate (BIT), but in a multiple-access OR-channel, the threshold is kept low to minimize the probability of mark errors (recording a space at times that should have been marks) at the expense of accepting potentially high space error rates (recording marks at times that should have been spaces). Specifically, the receiver's threshold is set sufficiently low so that, with extremely high probability, marks will be successfully detected whatever the jammer does even though doing so permits the jammer to easily force the receiver to record false marks. The channel thus embodies the notion of an “indelible mark” in that any mark transmitted is virtually guaranteed to be received. This is especially true if marks represent pseudorandom energy added to the RF spectrum.

If the channel’s marks are sufficiently indelible then messages can neither be altered nor removed. Jammers must therefore expend energy to flood the channel with additional messages. The receiver can use asymmetric means, such as digital signatures and certificates, to sift through the received messages identifying those that are valid; as long as the receiver has the processing capacity to handle all of the messages the attacker can insert within their energy budget, the channel remains available.

5. A brief primer on superimposed and concurrent codes

The notion of using multiple-access OR-channels to store multiple messages simultaneously is not new; the field of superimposed codes has arisen over the past half century to explore them (Kautz 1964). For our purposes, a simple example will reveal the key properties and weakness of traditional superimposed codes when applied to the problem of jam resistance.

Consider the vectors depicted in Figure 1. The bottom three are the definitions of the codewords for the messages ‘A’, ‘K’, and ‘Q’. The top is the received packet. Which, if any, of the three messages are contained in the packet?

Figure 1: Superimposed message packet and codewords.

Recalling that a codeword is contained in a packet if and only if all of the marks that define it are present, it is easy to determine that ‘A’ and ‘Q’ are in the packet but that ‘K’ is not. Understanding this much means understanding the heart of superimposed codes.

But does the packet contains other messages? The obvious approach is to examine each possible codeword against the packet. This is, in fact, the way that most superimposed codes are decoded. This is manageable if the total number of codewords in the codebook is sufficiently small which, with modern processors, could still mean billions of codewords. But if the number of codewords is exponentially large, this is simply not feasible.

In practice, superimposed codes are almost exclusively used in applications where “membership tests” are sufficient. A membership test is simply a query of whether a specific codeword is contained in a packet. In such applications it is never necessary to fully decode a packet. But to be of use in jam resistant communications the entire list of codewords (from an exponentially large codebook) contained in the packet must be produced. At least one paper in the recent literature (Cormode 2003) asserts that no efficient means of decoding arbitrary codewords from very large codebooks yet exists.

This is where the new theory of concurrent codes takes over. Simply put, a concurrent code is defined to be a superimposed code that is efficiently decodable. To illustrate this, Table 1 is the codebook for our previous example. The messages (second to last column) are simply the letters of the alphabet. The codeword for each message consists of marks at the locations listed in the first seven columns. For instance, the codeword for 'S' is the set of marks at locations {27, 19, 63, 46, 10, 58, 66}. The bit vector in the last column is simply the five-bit value assigned to the message plus two appended zeros. The purpose of the appended zeros, which act as checksum bits, will become clear later.

Table 1: Codebook for a 26-element concurrent code.

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	47	05	Y	1100000
				19	53	84	Z	1100100

The structure of the table reveals the key to efficient encoding and decoding; each possible message prefix is associated with a mark at a particular location in the codeword. For instance, any messages with the prefix '101', namely {'U', 'V', 'W', 'X'}, has a mark at locations 11. Since these messages also share prefixes '1' and '10' they also have marks at locations 27 and 19. The set of marks that define a codeword are thus functions of all possible prefixes of the message. If a hash function is used to translate an arbitrary prefix to a pseudorandom mark location then that hash function defines the codebook.

Decoding a packet uses essentially the same process by maintaining a list of possible message prefixes of increasing length. To illustrate this, let's decode the packet in Figure 1 using Table 2 to define our hash function. To begin we merely note that, if there are any messages at all, they must start with either a '0' or a '1'; we therefore start with a message list containing the 1-bit prefixes {0, 1}. We then eliminate from the list any prefixes that do not have marks at the locations indicated by the hash of that prefix. In this case, H(0)=36 and H(1)=27 are both present and hence nothing is eliminated. We now expand the message list by recognizing that the next bit in any message must be either a '0' or a '1', hence we replace each message prefix with two new prefixes, one having a '0' and the other having a '1' appended. In this case, our next message list would be {00, 01, 10, 11}. If we have reached the appended checksum bits we only replace each prefix with a new prefix having a '0' appended.

We repeat this process until our list is either empty, indicating that the packet contains no messages, or contains full length padded messages. The following are the message lists for each stage of decoding for this example:

1-bit: {0, 1}
2-bit: {00, 10, 11}
3-bit: {000, 100, 101, 110}
4-bit: {0000, 1000, 1100}
5-bit: {00000, 10000, 10001, 11000, 11001}
6-bit: {000000, 100000, 110000, 110010}
7-bit: {0000000, 1000000, 1100100}

If any full-length messages survive, we recover the messages by stripping the appended zeros from the end leaving us with {00000, 10000, 11001} which correspond to {'A', 'Q', 'Z'}

Had we not appended the zeros and only used the bits assigned to the message we would have ended up with two additional messages, namely '10001' and '11000', corresponding to the messages 'R' and 'Y'. These false messages are called hallucinations because they're messages that only appear real. Hallucinations arise as coincidental interactions between the marks in the packet. In the case of 'R', some of the marks were provided by message 'A' and the rest by message 'Q'. Thus, had we not appended zeros, anytime 'A' and 'Q' were both in the same packet the false message 'R' would also be produced by the decoder. In the case of 'Y', the one mark not provided by the message 'Z' is present due to random noise. Neither message became a hallucination because each appended zero bit acts as a checksum bit by placing an additional mark at a location that is a function of the entire message that is independent of any mark locations produced by any other message.

In addition to hallucinations that survive the entire decoding process, there are also hallucinations that arise during the decoding process and then expire at some later point. These are called working hallucinations. The significance of hallucinations and measures to suppress them will be discussed in Section 7.

6. The BBC encoding and decoding algorithms
The following are simply pseudocode restatements of the algorithms illustrated in the prior section.

6.1. BBC message encoding algorithm
To construct the n-bit codeword C for the m-bit message M containing k checksum bits using the hash function $H()$, perform the following:

1) Initialize all bits in C to zero (spaces).
2) Form the padded message M' by appending k '0' bits to M
3) For each of the (m+k) prefixes of the M', place a mark (i.e., '1') in C at location $H(M'(i))$ where $M'(i)$ is the i-bit prefix of M'.

6.2. BBC packet decoding algorithm
To decode all of the m-bit messages from a packet P, where each codeword is n-bits and was encoded with the BBC encoding algorithm using k checksum bits and a hash function $H()$, perform the following.

1) Begin with a message list containing all possible 1-bit prefixes (i.e., {0,1}).
2) UNTIL (message list is empty OR the prefixes are m+k bits long):
2.1) FOR (each *prefix* in the list)
2.1.1) IF (packet contains a mark at location H(*prefix*)):
2.1.1.1) IF (*prefix* length < m): Add a *prefix* to the list by appending a '1' bit.
2.1.1.2) IF (*prefix* length < m+k): Add a *prefix* to the list appending a '0' bit.
2.1.2) IF (*prefix* length is less than m+k): Remove the *prefix* from the list.
3) Strip k bits from the end of each *prefix* in the list.

The resulting message list, which may be empty, contains the messages whose codewords were contained in the packet.

Note that the above description is for a breadth-first search of the codebook. It is also possible to perform a depth-first search which permits the algorithm to run in a fixed amount of memory.

7. Expected Codec Workload

Although legitimate messages can be recovered even under significant jamming, the jammer still impacts the receiver since added mark can produce working hallucinations requiring additional work by the codec. The number of steady state working hallucinations is a function of the number of codewords in the packet, regardless of their source - the legitimate sender (M_S) or the attacker (M_A) - since each is a spawning source for hallucinations. It is also a function of the overall packet mark density (μ_P), since marks act as "food" for the hallucinations allowing new ones to be born or existing ones to survive. The number of working hallucinations per codeword (M_{HW}), as a function of packet mark density, is given in Equation 1.

$$M_{HW} = (M_S + M_A)\left(\frac{\mu_P}{1-2\mu_P}\right) \quad (1)$$

At a packet mark density of 50% the number of working hallucinations becomes infinite and the channel is jammed. However, at a packet mark density of 33% there is only working hallucination for each actual message. This is noteworthy since, if the attacker can commit this much energy, they can probably afford to jam the channel completely. Conversely, a more restrictive energy budget almost guarantees that the decoder can handle the working hallucinations if it can handle the actual message load.

The number of working hallucinations in steady state is also the number of hallucinations that are expected to exist after all of the non-checksum prefixes have been decoded. These terminal hallucinations are then suppressed by the checksum bits. Since each hallucination's probability of surviving the next checksum bit is equal to the packet mark density, the expected number of hallucinations for *k* checksum bits as a function of the packet mark density and the number of messages added by the attacker (assuming the attacker is injecting many more messages that the legitimate sender, who is likely only transmitting a single codeword) is given in Equation 2.

$$M_H = \left[M_A\left(\frac{\mu_P}{1-2\mu_P}\right)\right]\mu_P^k \quad (2)$$

The number of checksum bits needed to provide excellent protection against hallucinations is quite manageable. For instance, if the codec parameters are chosen such that an attacker must flood the receiver with one million messages in order to bring the packet mark density to the 33% value mentioned previously, there will be approximately one million terminal hallucinations. Just 16 checksum bits results in less than a 2.5% change of even a single hallucination surviving. This compares very favorably with the 32-bit checksums routinely used in communication protocols.

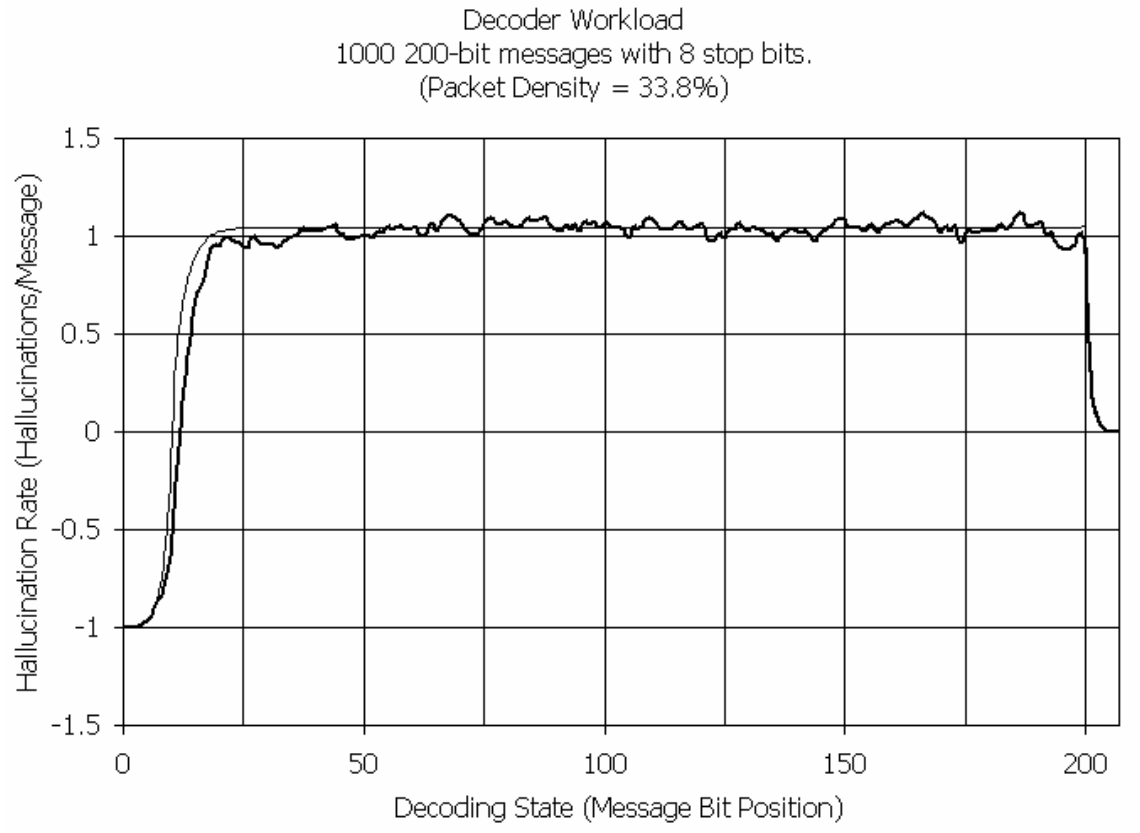

Figure 2: Hallucination performance of a concurrent codec.

The very tame and predictable hallucination behavior of a BBC-codec is illustrated in Figure 2. In this example, one thousand 200-bit messages were superimposed and just eight checksum bits were sufficient to exterminate all of the terminal hallucinations. The "negative" hallucinations at the beginning is an artifact of the fact that the total number of partial messages that can be represented by the limited number of message bits is fewer than the number of messages known to be in the packet.

Readers interested in a thorough discussion of the behavior of concurrent codes are referred to the original technical report (Baird 2007).

8. Demonstration Systems

Demonstration systems using visual, audio, and radio frequency transmissions have been constructed. The visual demo (Schweitzer 2007) represents each packet bit as a pixel in an image and provides a very intuitive impression of the potential power of concurrent codes. The audio demo represents each packet bit as a short chirp from a computer's sound card that permits observers to directly sense the overlapping packets as two computers transmit messages while a third records the sound in the room and decodes the resulting packets. The radio frequency demo uses software defined radios to simultaneously transmit message packets from multiple transmitters while a receiver captures and decodes them.

9. Conclusion

The reliance of traditional spread spectrum on symmetric keys presents a major roadblock to the successful implementation of the Global Information Grid as the scale of the networks collides with the practical scalability limits of symmetric key management schemes. The emerging theory of concurrent codes and the BBC codec algorithms derived from it offer solutions to this problem by permitting the development of jam resistant omnidirectional links that do not rely on shared secrets.

Acknowledgements

This work was sponsored in part by the Air Force Information Operations Center (AFIOC),

Lackland AFB, TX, and was performed at the Academy Center for Information Security (ACIS) at the United States Air Force Academy.

References

Alberts, D. (2002) Information age transformation: getting to a 21st century military, Command and Control Research Program (CCRP) Publication Series.

Alberts, D., Garstka, J. and Stein, F. (2000) Network Centric Warfare: Developing and Leveraging Information Superiority, 2nd Edition (revised), Command and Control Research Program (CCRP) Publication Series.

ASW (1999) "Premises Challenged of 'Sole Means' Satellite Navigation Risk Assessment Study", Air Safety Week, 22 February, http://findarticles.com/p/articles/mi_m0UBT/is_8_13/ai_53951154.

Baird, L, Bahn, W. and Collins, M. (2007) "Jam-resistant communication without shared secrets through the use of concurrent codes", U.S. Air Force Academy Technical Report USAFA-TR-2007-01, 14 February.

Bahn, W., Baird, L. and Collins, M. (2007) "Impediments to systems thinking: Communities separated by a common language", *Proceedings of the 4th International Conference on Cybernetics, Information Technologies, Systems and Applications (CITSA), (III)*, pp 122-127, 12-15 July, Orlando, Florida.

Brewin, Bob (1998) "Rogue transmitter knocks out GPS signals", Federal Computer Week, 12 April, http://www.fcw.com/print/4_28/news/66029-1.html.

Carroll, J., Van Dyke, K., Kraemer, J. and Rodgers, C. (2001) "Vulnerability Assessment of the U.S. Transportation Infrastructure that Relies on GPS." *ION National Technical Meeting*, Long Beach, CA, 22-24 January.

Cormode, G. and Muthukrishnan, S. (2003) "What's hot and what's not: Tracking most Frequent Items Dynamically", *Proceedings of ACM Principles of Database Design*, pp 296-306.

FAA (Federal Aviation Administration) (1996) *1996 Federal Radionavigation Plan*, U.S. Gov't Printing Office, July.

FAA (Federal Aviation Administration) (2005) *2005 Federal Radionavigation Plan*, National Technical Information Service, Springfield, VA 22161, July.

Foster, J. and Welch, L. (2000) "The Evolving Battlefield", *Physics Today*, Vol 53, No. 12, December, p 31.

GAO (2004) "Defense Acquisitions - The Global Information Grid and Challenges Facing its Implementation", U.S. Government Accountability Office Publication GAO-04-858, July, http://www.gao.gov/new.items/d04858.pdf.

Gowens, J. and Young, J. K. (2000) "FY2001 Annual Report of the Communications and Networks Consortium", Army Research Laboratory Collaborative Technology Alliance Program.

Kautz, W. and Singleton, R. (1964) "Nonrandom Binary Superimposed Codes", *IEEE Transactions on Information Theory*, Vol 10, pp 363-377.

NSA (2008) "The GIG Vision, Enabled by Information Assurance", [online], National Security Agency, http://www.nsa.gov/ia/industry/gig.cfm.

Peterson, R., Ziemer, R., and Borth, D. (1995) Introduction to Spread Spectrum Communications, Prentice Hall, Upper Saddle River, New Jersey.

Sutton, Oliver (1998) "FAA drops GPS bombshell", Interavia Business and Technology, September 1998, http://findarticles.com/p/articles/mi_hb3126/is_199809/ai_n7792845.

Forouzan, B (2008) Cryptography and Network Security, McGraw Hill, New York, New York.

Schweitzer, D., Baird, L. and Bahn, W. (2007) "Visually Understanding Jam Resistant Communication", *Proceedings of the 3rd International Workshop on Visualization for Computer Security (VizSec)*, 29 October.