Skepticism and Cryptography

Barry S. Fagin
Leemon C. Baird
Jeffrey W. Humphries
Dino L. Schweitzer

Academy Center for Information Security
Department of Computer Science
2354 Fairchild Drive
US Air Force Academy, CO 80840

ABSTRACT

Cryptography is an essential component of America’s national security infrastructure. Billions of dollars are spent on cryptosystems every year, in both the public and private sector. Unfortunately, the field is rife with dubious claims, snake oil salesmen, and outright fraud.

This paper highlights the importance of skepticism and critical thinking in the role of evaluating and procuring cryptosystems. We discuss our experiences in teaching future leaders about testing extraordinary cryptographic claims by asking hard questions, and show examples from our own experience. We believe that the rigorous application of skepticism and critical thinking in cryptography are absolutely essential to the wise use of America’s resources and the security of the nation.

INTRODUCTION

Cryptography is an essential component of America's national security infrastructure. Companies require it to keep their assets secure from economic espionage. The military requires it to keep operations secret from adversaries. The Global War on Terror cannot be fought without it.

Because of its role in protecting property, information, and lives, cryptography is big business. Several years ago, Bruce Schneier estimated the private market for global crypto to be over $2 billion [Sc94a]. It is surely much higher now. Last year, one crypto vendor alone reported annual earnings of over $300 million [RSA05]. Currently, one consulting group estimates the market for email security, which is heavily based on cryptography, to hit $6 billion by 2010 [Ra06]. If we include expenditures for Information Technology Security, which also relies heavily on cryptography, we obtain figures well over $10 billion [AMI06].

Public sector figures are more difficult to calculate, since such expenditure is often classified. Still, the importance of cryptography in both governmental and defense expenditures is difficult to overstate. In June of 2006, in response to concerns about data losses, the Bush administration issued guidelines requiring the encryption of all "sensitive" data on every laptop or handheld device in the federal government [Kr06]. This is an important but massively expensive undertaking. The Baltimore Sun, in a recent article on hacker attacks against the Department of Defense, estimated the costs of the Public Key Infrastructure initiative at over $2 billion [Go06]. This is only one security program in one agency of the US Government.

It seems clear that billions of public and private dollars are spent every year on products built on cryptography. The national information infrastructure cannot be secured without it. The computer security industry is growing by leaps and bounds, and information security is arguably the single most active research area in computer science. The explosive growth in this area and the resources involved are nothing short of astonishing.

Unfortunately, the mathematics involved in most cryptosystems are not accessible to non-specialists, and it is typically non-specialists (CEOs in industry, generals in the military, and senior civil servants in the public sector) who make big purchasing decisions. This combination of big bucks and little know-how attracts a predictable assortment of con artists, snake oil salesmen, and well meaning but naïve investors who genuinely believe the claims of their marketing literature. If we were simply talking about dangerous cooking gadgets or suspicious fat-burning products, we might simply refer the matter to the Federal Trade Commission. But cryptography will become an essential part of the economic and security infrastructure of every industrialized nation on the planet. Getting it right is absolutely vital.

The way to get it right is with skepticism and critical thinking. We believe that anyone involved with the evaluation, purchasing, production, installation and maintenance of cryptosystems must demand extraordinary evidence for extraordinary claims, must know how to distinguish truth from nonsense, and must not be fooled by jargon. We offer our insights as to how this might be accomplished.

We begin with a high-level overview of cryptography, to introduce the terminology and explain what the issues are. We show what distinguishes good crypto from bad, we discuss the social factors leading to dubious crypto claims, and show features of a typical snake-oil sales job. We then show a typical procurement process associated with the purchase of a cryptosystem, and identify points of vulnerability to bad science and false claims.

Here at the United States Air Force Academy (USAFA) we teach a course in cryptography for computer science majors, part of which is devoted to dealing with cryptographic hokum. We share some insights from our experiences, and show some actual examples of what our students may eventually have to deal with. We conclude with a standard set of questions and guidelines for evaluating cryptographic claims, and a call for required work in skepticism and critical thinking for anyone involved in cryptography.

OVERVIEW OF CRYPTOGRAPHY

Although there is a wide variation among key-based cryptographic techniques, they all fall into one of two categories.

Figure 1 is derived from Schneier [Sc94b]. It shows an older and simpler type of key-based cryptosystem. The *plaintext* message (as originally written by the user) is *encrypted* in some way to produce *ciphertext.* The box labeled "Encryption" implements an *encryption algorithm,* a step-by-step procedure that transforms the human-readable plaintext message into the encrypted ciphertext version. This algorithm requires a *key* in order to function correctly. The same key is used in the *decryption algorithm,* which transforms the ciphertext back to the plaintext. This type of cryptosystem is called a *symmetric* cryptosystem, because encryption and decryption use the same key.

For example, a simple shift cipher that replaces every letter in the plaintext with the letter three ahead of it in the alphabet[1] would qualify as a symmetric cryptosystem. The plaintext "ATTACK" would encrypt to "DWWDFN". Encryption is performed by adding the key to each letter, decryption by subtracting. In this case the number 3 would be the key. Julius Caesar is said to have used a system like this to communicate with his generals. This system may have been effective two millennia ago, when not much was known about cryptography, but in modern times a Caesar cipher is not very secure. We will see the reasons shortly.

It is normally assumed that the encryption and decryption algorithms are known to adversaries. It is the key that is assumed to be secret, and on which the security of the cryptosystem depends. Keys for symmetric cryptosystems have included codebooks, tape with holes punched in it, and in ancient times a message printed on parchment and

[1] Letters near the end of the alphabet would "wrap around": Y would encrypt to B, for example.

wrapped around a cylinder. Modern cryptosystems use numbers for keys, typically chosen to be very large. Throughout this article, we will assume that all keys are numbers.

Symmetric cryptosystems require two trusted individuals who wish to communicate to agree on a key and then keep it secret. Without such secrecy, the system is not safe to use. Recent developments in cryptography have produced improved *asymmetric* cryptosystems, as shown in Figure 2.

Asymmetric cryptosystems use different keys for encryption and decryption. Each user chooses his or her own two keys, using specially designed computer software. These keys are chosen in such a way that there is no known way to easily calculate one from the other without using information known only to the user. This means that one key can be made public, and in fact it is advantageous to do so. For example, if I use a special computer program to pick my own personal encryption and decryption keys, I can make my encryption key public but keep my decryption key secret. My encryption key can then be used by anyone to send me secret messages that only I can read. This is known as *public-key* cryptography.

GOOD VS BAD CRYPTO

What is the goal of cryptography, and how can we distinguish good crypto from bad? We start with some definitions [Sc94b].

Secure cryptosystems are those for which an adversary cannot recover the plaintext from the ciphertext without expending resources (typically computer time or brainpower) that exceed the value of the plaintext. Note that it is generally assumed that an adversary has access to the ciphertext as well as the encryption and decryption methods used. After all, communication channels can be eavesdropped upon, and coding machines can be captured. In World War II, for example, the Allies intercepted numerous encrypted German transmissions, and eventually captured the encryption and decryption boxes known as “Enigma” machines[2].

We can go even further and say that a cryptosystem is *unconditionally secure* if no matter how much ciphertext an adversary has, there is no way to recover the plaintext. There is actually a system that has this property, but because it requires keys as long as the messages you need to transmit it is difficult to implement and impractical to use. Additionally, because you have to use a different key each time, this system is of interest only for a narrow range of applications.

Instead, we are typically more concerned with *computationally secure* or *strong* cryptosystems, which can’t be broken with current or future technology. Clearly, the question of what constitutes future technology makes this definition somewhat subjective. *Weak* crypto can be broken by cryptographers, existing computers, or soon-

[2] One such capture was dramatized in the 2000 film “U-571”.

to-exist computers, in a reasonable amount of time. Again, the question of what is reasonable makes the definition somewhat subjective. It is safe to say, however, that the strongest cryptosystems are those that cannot be broken with any known technology or with any technology envisioned for several years into the future. As Bruce Schneier notes [Sc94b], "I would consider an algorithm that takes a billion times the age of the universe to break to be computationally secure."

Note that for the cryptographic community to have high confidence that a cryptosystem is strong, the algorithms involved must be submitted to public scrutiny. The strongest cryptosystems known are based on published algorithms that have been subject to strenuous analysis by the cryptographic community. Such analysis enables us to provide a mathematical foundation for security claims. While the current foundation remains less rigorous than we would like, it is the best strategy we have for ensuring security. Cryptosystems that rely on mysterious "black boxes" or "secret technology" for their encryption and decryption algorithms should never be considered secure. We will have more to say about this shortly.

It is interesting to note that all strong cryptography known today rests on the *assumed* difficulty of certain mathematical problems (for example, the difficulty in quickly factoring a number that is the product of two large primes). At the moment, we have high confidence in those assumptions based on these problems' persistent resistance to solution despite the best efforts of mathematicians over the centuries. However, should these problems be solved, much of our cryptography will no longer be secure.

Similarly, strong crypto also rests on the *assumed* rate of technological advance in computing power. Computing power goes up by a factor of 4 every three years, so it is tempting to think that today's secure crypto might become weak tomorrow. However, in the same way that adding a single digit onto a number gives you ten times as many numbers to work with, adding a single bit to a key makes it twice as hard to guess. This means that going from a 32-bit to a 64-bit key in a strong cryptosystem, for example, makes the system 4 billion times stronger. This provides crypto designers with a reasonable barrier of security against advancing compute power for the foreseeable future. However, should radical ideas like quantum computing [NiCh00] become a reality, then the foundations of secure cryptography could crumble. For just this reason, quantum computing is an extremely active area of research around the world.

WHERE DUBIOUS CRYPTO CLAIMS COME FROM

The prevalence of dubious cryptographic claims arises from the interaction of several factors. Any one or two of them separately would not amount to much, but when combined they wreak havoc. We see the following social conditions as contributing to the unique situation in the cryptographic industry:

1) The mathematical sophistication required to understand modern cryptosystems. Cryptography does not require calculus or knowledge of continuous functions, those branches of general mathematics considered among the most difficult. It does, however, deal with binary numbers instead of the more familiar decimal system, it uses numbers much larger than ordinary experience, and it manipulates them in a counterintuitive way.

The number system we’re most familiar with is based on the number ten[3]. This system uses ten symbols to indicate value: The standard Arabic numerals 0,1,… 9. The decimal system writes symbols in particular places to designate specific values: the ‘9’ in ‘963’ means “nine hundred”, while the ‘9’ in ‘29’ means “nine”. Digits are multiplied by different powers of ten in a number, depending on the position where they appear.

Binary numbers work the same way, it’s just , as we often tell students, computers have two fingers. The binary system is based on the number two and therefore needs only 0 and 1, the first two symbols of our number system. Like the decimal system, it is positional. We can write the number of letters in the alphabet as 26 if we use decimal numbers, or we can write it as 11010 if we write it in binary. It’s just a question of which convention we want use. Most cryptographers and computer professionals are comfortable working in either number system, depending on the context. Most of the rest of the world is not.

We can get a feel for the magnitude of the numbers involved by considering how many different values a number can take on based on the number of digits it has. A single digit number in base ten can take on ten values: 0 through 9. A two-digit number can take on a hundred values: 00 through 99. The number of values is always ten to the number of digits, a function that gets large very quickly.

The same is true for binary numbers. A cryptographic key that is 256 bits long, for example, could take on one of 2^{256} possibilities. This is a number with 78 decimal digits, on the order of the estimated number of atoms in the universe [Dy79]. Cryptographers normally speak of the number of bits in a key, not the number of possible keys, but the door is left open for a hustler to say things about mind-bogglingly large numbers in hushed and mystical tones.

Nor are the mathematical operations employed on these numbers in the usual fashion. Arithmetic in cryptography is normally done within a *field,* a special type of mathematical structure. One example of work in a field involves multiplying two numbers together in the traditional fashion, but then taking the remainder when divided by another number as the final answer. This is known as a *modulus* operation, or mod for short. For example, in the field of integers mod 13, 5x4 = 7, not 20, because all operations are based on the remainder when divided by 13. Cryptographers do this type of math all the time.

Most of us have enough exposure to cars, TVs, and real estate that we can recognize when someone is saying things about them that don’t make sense. Not so with

[3] Our use of ten is merely an accident of biology. Knowing that you can count on your fellow humans to have two appendages with five fingers each makes for a convenient system we can all rely on. It’s no coincidence that another word for ‘finger’ is ‘digit’.

cryptosystems. The numbers involved and the way they are used are utterly divorced from most people’s daily experience. This makes detecting nonsense much harder.

2) Lack of scientific training among those with decision authority. The mathematical sophistication required to understand crypto doesn’t by itself prevent the detection of nonsense. There are plenty of technically trained professionals in the world who can separate truth from falsity. The problem is that technical personnel are not always among those with decision authority. Senior management in corporations or generals in the military are typically non-specialists who achieved their positions based on leadership skills and willingness to make tough decisions under stress. While not unheard of, it is unusual to find practicing mathematicians or engineers at high levels of management in large organizations. The stereotypes of the engineer with no social skills and the manager who can’t do math, so skillfully lampooned in popular culture [Ad97], are all too well grounded in reality.

This means those who make the decisions and sign the checks to purchase millions of dollars worth of cryptographic systems are easily vulnerable to jargon-laden salesmen and piles of papers with impressive-sounding academic citations. Unless their technical people are familiar with cryptography *and brought into the decision process*, senior management and military leadership may be at risk for investing resources in inferior or even worthless products.

3) Imperfections of the marketplace. In a competitive market, over time, sense wins out over nonsense and good products drive out bad ones. The questions of “How competitive?” and “How much time?” lie at the heart of bad crypto, and indeed bad products in general.

It is a regretable fact of human economic activity that, to make money, one does not have to provide value. One simply has to convince others that a product has value. If the founders of a startup can convince investors of the value of their product, they can amass large paper fortunes quickly. If the owner of a computer security firm can convince a buyer that his product is the latest thing, he can cash out before the product is suitably investigated by knowledgeable professionals in the marketplace.

Markets for corporate finance have developed mechanisms for preventing these sorts of transactions. Reputable venture capitalists hire technical experts to help them with “due diligence” requirements before investing. Successful investors didn’t get that way by being reckless with their capital. Nonetheless, the fact that convincing others of value can substitute for true value in the marketplace will always be an issue with cryptosystems, particularly given their unique requirements.

4) Creation of conditions friendly to self-deception

This is not to say that all cryptographic claims are deliberately fraudulent. Inventors and vendors may genuinely believe their claims.

When belief systems reinforce positive things about ourselves, we are more likely to subscribe to them. People associated with questionable crypto may genuinely believe nonsense because such beliefs can make them wealthy. All good sales professionals know the importance of belief in the product and a positive attitude. It would be wonderful if product X really were superior to everything else on the market, cost one tenth as much, and would keep America secure, especially if you've got stock options at the company that makes it. Under those circumstances, surrounded by peers and co-workers in similar circumstances, even normally cautious people might let down their guard.

The importance of information security and the large sums of money involved create circumstances in which self-deception can spread like wildfire. Bad crypto just drops the match.

5) Secrecy and national security

The previous factors are all important in understanding why so much bad crypto is out there. In our opinion, however, the biggest enablers of deceit in cryptography are the need for secrecy and protection of national security.

Cryptography is used to keep things secret. More accurately, it combines laws of mathematics with knowledge possessed by trusted individuals to deny information to non-trusted individuals. Non-trusted individuals may be persons from whom we merely wish to keep information private. They may be economic competitors, or political opponents. In all cases, secrecy is part and parcel of the cryptographic enterprise.

The importance of secrecy becomes all-encompassing when national security is at issue. In this case, the stakes are far higher than having your grocery-buying habits become public or a company losing its competitive edge. People die and nations fall when the right information winds up in the wrong hands.

But secrecy and mystery are antithetical to scientific truth. The best way to determine the validity of a scientific conjecture or the soundness of a commercial product is through open processes like publication, verification, discussion, and analysis. Because of cryptography's use in maintaining secrecy, it is often quite easy to avoid these processes in the name of "secret algorithms" or "national security".

In fact, there is no reason to keep the details of a cryptographic algorithm secret *if the security of the algorithm is the only concern.* The best cryptographic systems in use today rely on fundamental properties of mathematics for their security, and their details are completely public. Open algorithms are rigorously tested by the cryptographic community. If they pass muster, they enter into the marketplace with a high social degree of confidence in their soundness.

Cryptographic snake oil, by contrast, often insists that its algorithms be kept secret for reasons of "national security", or that users buy expensive product installations and then sign a sprawling non-disclosure agreement. The only reason for a company to keep a cryptographic algorithm secret is to make money. If a system relies on secrecy of its inner workings to maintain security, you can bet there’s something fishy going on.

FEATURES OF BOGUS CRYPTOGRAPHIC PRODUCTS

There are many vendors who try to sell cryptographic products that are insecure or lack the accepted evidences of security. The following buzzwords and claims often indicate the presence of snake oil. Some of these were first pointed out by Schneier in [Sc94c], others are based on our personal experience.

A cipher that is new, proprietary, or patented

As explained in the last section, the best guarantee of security is the repeated subjection of ciphers to analysis by the cryptographic community. New ciphers are thus unlikely to be secure. Furthermore, even if one were to be secure, there is no way that the world could know that. Ciphers, like fine wines, are better when they are well aged.

More secure than existing ciphers

The ciphers currently in use do not have known security flaws. Otherwise they would not be in use. If a vendor claims that current ciphers are insecure, that is a suspicious claim. If the vendor found a way to break existing ciphers, then that break should be peer reviewed and published. Vague claims of insecurity without proof are simply snake oil.

Unusual or revolutionary math

If a system boasts new or revolutionary math, that often indicates it hasn't yet been studied sufficiently by the cryptology community. “Chaos theory”, “3D matrices”, and other buzzwords should set off a skeptic’s warning bells. It does take some knowledge of standard practice to distinguish standard math from novel math. For example, the phrase “elliptic curves” might appear in a snake oil brochure, even though elliptic curves are commonly used to improve cryptosystems.

Giant keys

Snake oil vendors often advertise absurdly long keys, such as keys with a million bits. Standard symmetric ciphers use keys up to a couple hundred bits, and asymmetric ciphers use keys up to a couple thousand bits. These are sufficient to prevent a brute force attack where the attacker tries all possible keys. If the concern is brute force, then these keys are enough. If the concern is a more intelligent attack, then there's no reason to think longer keys are more secure. Therefore, enormous keys are a sign that the vendor doesn't understand cryptography, or hopes that you don’t.

One Time Pads (OTPs) or pseudo-OTPs

As mentioned earlier, there is in fact a system that is truly unbreakable, known as the One Time Pad (OTP). Unfortunately, it requires that the secret key be 1) truly random, 2) truly secret, 3) as long as the message, and 4) shared by both parties. Such a system is too unwieldy to be useful, except in the most extreme circumstances. Snake oil vendors often build simple stream ciphers and claim that they are unbreakable OTPs, even though they clearly are not and so do not have the perfect security of a true OTP. The variant term "pseudo-OTP" is an even stronger indication of snake oil.

Unbreakable! Provably Secure!

No cipher is unbreakable except an OTP, and true OTPs are unwieldy. Cryptographers often publish security proofs for algorithms and protocols, but they always rest on unproven assumptions, and so do not actually prove that the systems are secure. Such security claims in marketing literature are often signs of snake oil.

NSA-tested! Certified! Military-grade security! Designed by NSA employees!

The National Security Agency does not test new ciphers for private companies and release the results. Having ex-NSA employees involved in a project hardly guarantees that it will be secure. Various types of security certification are available, but they generally attest that the software correctly implements the cipher, rather than checking that the cipher itself is secure. These exaggerated claims are often used by snake oil vendors.

Experts couldn't break it – No one claimed a contest prize

Novice cryptographers often email new ciphers to experts, or post new ciphers to newsgroups such as sci.crypt. Unless the algorithm is well described and interesting in some way, it is usually ignored. If this silence is interpreted as an endorsement, that is a sure sign of snake oil. A similar problem is a "contest" where someone offers $100 to break some new cipher. Professionals are rarely interested in such contests, because the prize is too small to justify the time required, and because a break of such a cipher is unlikely to be publishable in a peer-reviewed journal or conference.

Secret Algorithm

If an algorithm or protocol is kept secret, that doesn't *increase* the security, it *decreases* it. If an algorithm is used in widely-distributed software, then an attacker will be able to decompile the software and extract the algorithm. Secrecy doesn't help. The only effect of secrecy is to prevent other security experts from finding problems or verifying the security. Snake oil vendors often keep their cipher algorithms secret, not realizing that the effect is the opposite of what they desire.

Appeals to the fallacy of exhaustive search

One suspicious sign in a crypto pitch is a focus on how long it would take to try all possible keys. This is what is known as an *exhaustive attack.* This fallacy is so common that it is worth going into in some detail.

Exhaustive attacks are guaranteed to work, but only if you have enough time. If the number of possible keys is large enough, it could take millions of years to try them all. A published statement to that effect from a credentialed researcher will make a nice marketing sound bite: “Dr. I.M. Verysmart at Prestigious University publishes paper proving SnakeOilCipher™ takes a million years to crack!”[4] Sadly, we would not be surprised to see Dr. Verysmart on SnakeOilCipher’s Board of Directors.

[4] There is, as of this writing, no such company as SnakeOilCipher

The problem is that while exhaustive attacks can indeed require enormous amounts of time, they don’t have much to do with how secure the system is. There are lots of ways to break bad cryptosystems without trying all possible keys. Anyone who’s ever done the ”cryptoquotes” on the puzzle page of the newspaper knows this firsthand.

Cryptoquotes are simple substitution ciphers; one letter is substituted for another everywhere it appears. The list of what letter substitutes for what is the key. For example, the plaintext

CRYPTOQUIZZES ARE SIMPLE SUBSTITUTION CIPHERS

could be encrypted to the ciphertext:

QWERTYUIOPPAS DWA SOFRGA SIHSTOTITOYJ QORKAWS

using the key shown in Table 1.

How many different ways are there to generate a cryptoquiz? This is the same as asking how many possible keys could have been used.

It is not difficult to work this out. This first letter could have been encrypted in 26 possible ways, the next in 25, and so on until you get to the last letter and only one way to encrypt it.

This means there are 26*25*24*…*3*2*1 keys, written in mathematics as 26! This number is about 400 septillion, a 4 followed by 26 zeros. It’s rather large.

But of course cryptoquiz solvers don’t waste their time trying all possible keys until they come up with something that looks like a Mark Twain quote. They look for common patterns, assume that common letters in the ciphertext correspond to common letters in the plaintext (so they’re more likely to be E and T as opposed to X and Z), and rely on their knowledge of English to make good guesses at unknown letters.

This works because substitution ciphers are good entertainment, but bad crypto. They don’t rely on mathematical fundamentals for security, which renders them vulnerable to attacks like those above. An adversary (in this case the puzzle solver enjoying a morning coffee) has other ways to discover the key without trying all the possibilities. The fact that a cryptosystem has a huge number of possible keys has nothing to do with whether or not you should trust it to secure your information from prying eyes.

FIGHTING BACK IN THE CLASSROOM

Given the issues presented so far, what can be done? For us, as computer science faculty at the US Air Force Academy, our marching orders seem clear . We spend a lot of time developing critical thinking skills in our students, preparing them to ask hard questions and be suspicious of jargon-laden sales pitches.

The computer science curriculum at the Air Force Academy is a nationally accredited program that provides a broad introduction of the field to our graduates. Since most of the graduates with a CS degree enter the Air Force communications/computer career field, our program emphasizes some areas unique to their future professions. In particular, we offer a *concentration in information assurance* under the auspices of the Academy’s Center for Information Security that ensures our graduates are equipped to deal with the increasingly important issues of computer/network security and defense.

The Academy Center for Information Security (ACIS) is a research organization at the Air Force Academy established in the Fall of 2004 to promote the advancement of Air Force and DoD information superiority. ACIS fulfills two roles: research and education in information security. ACIS conducts basic and applied research in the areas of information security and works closely with Academy faculty and students on collaborative research projects. ACIS also develops, supports and coordinates improvements in education and training in the areas of information assurance and computer security. ACIS played a pivotal role in developing the concentration in information assurance within the computer science major.

USAFA’s concentration in information assurance consists of three courses in addition to those required for the bachelor of science in CS. The concentration includes courses in secure networks, computer security and cryptography. The goal is to provide the Air Force with more officers who are knowledgeable and skilled in the area of computer security and information assurance for the defense of the nation.

The Air Force, like most other major organizations, is constantly presented with “new” and ostensibly better solutions to the security problem in the information domain. One of our major objectives is to graduate individuals who can correctly understand, evaluate, and appraise the security claims made by these ever-present purveyors of security products and explain to decision makers why these products should or should not be pursued. It is absolutely critical, in terms of manpower, money, and security, that organizations employ people who can separate the euphemistic “wheat from the chaff”. To that end, we have developed a “snake oil” unit of instruction as a capstone lesson in our cryptography course that attempts to develop such discerning individuals. It may be helpful to first explain the overall goal of the cryptography segment of the concentration in order to motivate the efficacy of our “snake oil” unit of instruction.

The primary goal of the cryptography course is to prepare our graduates to understand current cryptographic concepts by combining critical thinking with the necessary mathematical background to appreciate what cryptography can and can’t do. We aren’t training professional cryptographers or cryptanalysts; rather, we want our graduates to value and understand the proper role of cryptography and to be able to spot “bad crypto” when they see it.

In order to do that, we spend the majority of the course pursuing eight major objectives. By the end of the course, students should be able to:

1. Explain, implement, and make use of the commonly used forms of cryptography, including both public key and symmetric key algorithms.
2. Understand and make practical use of the common types of cryptosystems, and understand comprehensively the principal advantages and vulnerabilities of each.
3. Understand the theoretical foundations and practical implementations of secret-key and public-key cryptographic systems.
4. Explain the theoretical problems as well as the practical issues associated with pseudo-random number generators, cryptographic hash functions, key control, key distribution, and key exchange algorithms.
5. Compare, contrast, and select the appropriate cryptographic techniques for a given security application and security policy.
6. Articulate a reasoned, well-thought-out position on the major public policy issues related to cryptographic technology.
7. Analyze appropriate mathematical problems in a form suitable for programming.
8. Implement cryptographic algorithms from their specifications by constructing programs in a modern programming language.

Several major themes keep recurring as we discuss each new topic. First, we want to emphasize critical thinking. Students are not presented with simple “plug and chug” problems. Instead, we provide them with never-before-seen (to them) crypto systems and ask them to make observations and suggest methods of breaking new codes. We ask them to suggest appropriate key sizes for a variety of problems and to justify their answers. We ask them to compare and contrast crypto systems and suggest which would be more appropriate for a given problem.

Second, we want our students to have an appreciation for the orders of magnitude that are involved in time and space with modern cryptographic systems. Most students new to cryptography have no appreciation of the key sizes, memory requirements, or computing time necessary to implement modern crypto systems. We have found that the factoring problem provides a good avenue to gain that appreciation. We require our students to implement a solution to the factoring problem using several sophisticated algorithms. Each student times his program and competes against their classmates to see who can factor number of certain magnitudes the fastest.

Third, we want our students to understand the nature of information. We spend at least ten percent of the course discussing entropy, Shannon’s theory of information, and randomness. These concepts are foundational to an overall understanding of what cryptography can do (and perhaps more importantly, what it can’t).

These themes run throughout the course as we discuss topics including classical crypto systems, basic number theory, modern symmetric and asymmetric systems, secret sharing, digital signatures, and hash functions. It's only after the foundational topics have been covered do we feel we can ask our students to synthesize and analyze new systems and evaluate them on their merits.

The course's final topic addresses the critical thinking and "snake oil" question. We use performance-based assessment (using the tools of critical thinking we have talked about all semester) to see how well our students can evaluate new products. During one final class session, we have the students pretend they work for a major organization and are asked to listen to a sales pitch of several new crypto systems. The instructors role play vendors and present their products to the "technical experts" (our students). After each product, we leave the role playing mode temporarily to ask our students to evaluate what they have just seen and present a recommendation to their "boss" (another role playing instructor). We, in turn, evaluate our students on how well they assessed the product we just pitched and suggest questions and areas of improvement. We then enter role-playing mode again, and present a new product, again allowing time to assess our students' abilities to ask good questions and spot "good" crypto from "bad".

We have heard nothing but good reviews on this approach to educating our graduates on what they may face in the future. Several graduates have given us feedback some years after the fact that the role-playing lesson was one of the best things they had seen to help them face their current job requirements. One graduate commented that what was "pretend back then is very real now".

CASE STUDIES

Much of what we teach in class is based on real-world snake oil examples, of which we have plenty to choose from. Cadets examine different products or sales pitches and are asked to identify potential problems. We show a few examples below.

PRODUCT A

Product A is an email security product. Here are some talking points from their marketing literature:

- **No one can ever decrypt your email. NOT EVER!**
- Much more secure than common algorithms such as DES, RSA, AES, PGP and Blowfish!
- These all can be attacked given time and money!
- Our product is verifiably, 100% secure – proven fact!
- Because our system is not based on an algorithm, there is no key to crack!
- Do you need a truly ***secure***, ***secure*** **FOREVER**, way to send sensitive files and data sheets?
- There is only ONE unbreakable encryption technique and now we can offer this to you.

- Our product is a point-to-point, person-to-person, *One-Time-Pad* encryption system based on uncrackable random numbers derived from **unpredictable thermal** and **‘shot’ amplifier noise**.

We’re not sure where to begin with this one. No one can decrypt your email? Not even you? What if you give your key to someone else? Note the claims of greater security than public and well-tested cryptosystems, the use of one time pads, and jargon. “Not based on an algorithm”? What in the world does that mean?

PRODUCT B

Here are a couple of clips taken from one security company’s web site:

> From the mathematical point of view, [our] algorithm is intuitively natural and less cumbersome to use than methods that are number-theory based. The algorithm utilizes the knowledge of higher dimensional affine spaces, and is based on the calculations of concrete polynomials. Moreover, it has the novel functions of error detection and master key.

The site goes on to report

> “… an encryption speed of 18 million bit *(sic)* per second and a decryption speed of 50 million bit per second [which] is much faster than the speeds of the secret key "triple DES 56" which are 6 million bit per second for both encryption and decryption. The complexity of [our algorithm] is (2^{90}) according to newest attacking schemes. This complexity is much higher than the commonly used criterion of (2^{80}) for the so-called "strong" cryptographic system.

This is, in our view, a classic case of snake oil. Note the buzzwords like “higher dimensional affine spaces” and “concrete polynomials”. It claims novelty, better performance than existing and well-known cryptographic standards, and uses the notion of complexity incorrectly. It’s hard to know exactly what they mean by the last sentence, but we suspect they expect customers to fall for the fallacy of exhaustive search.

PRODUCT C

Below is a proposal received by one of the authors in July of 2005:

> “ … We have a desire to have our encryption technology properly characterized by an algorithm that describes the number of combinations required to test all possible solutions. Normally this is a fairly simple exponential relationship for most encryptions based upon the size of the key. However, our encryption involves nine keys, each of variable length of up to one million bits, as well as a three-dimensional matrix - for

> starters. So the complexity of the mathematical description is not so easy. As I mentioned, we don't want to expose the entire nature of the process to the civilian academia [sic] due to the potential national security implications …”

This paragraph displays many of the key characteristics we have cited previously, including the use of jargon, large numbers, and the stated need for secrecy due to “national security implications”.

Note the immediate emphasis on exhaustive attacks to break the system. The “number of combinations required to test all possible solutions” would of course be exponential, and probably a number so unimaginably huge that it would make a terrific sound bite. Not that it would have much to do with security. Once again, we have the fallacy of exhaustive attack.

Product C’s description also emphasizes “nine keys”, which at first glance would appear to be nine time as good as a mere single-key system. But mathematically, it doesn’t make any difference. One key of a thousand bits, or ten keys of a hundred bits each, it’s all the same cryptographically. The total number of bits involved in all the keys is all that matters. Emphasizing multiple keys suggests at best mathematical ignorance, and at worst deliberate deception.

CONCLUSIONS

Cryptography and information security are multibillion-dollar industries. The economy of the modern world and the defense of every industrialized nation cannot be carried on without it. Unfortunately, the sophistication of the mathematics and the large amounts of money at stake combine to fill the field with snake oil salesman and bogus claims.

Fortunately, progress is being made. The National Institute of Standards and Technology has joined with the NSA to form a “Common Criteria” process [CC05] for increasing the confidence in cryptographic and information security related products. Recognizing the importance of improving information security training, the Department of Defense (DoD) has enacted policy directives requiring certification of Information Assurance (IA) professionals within the DoD, as well as basic IA training for all DoD employees.

But these types of advances will never be as effective as having the correct mathematical understanding of cryptography, a knowledge of what questions to ask, and most importantly a willingness to go to people who have both when a key decision needs to be made. When it comes to cryptographic technology, senior executives, government officials and policymakers all need to ensure that people with the appropriate technical skills and critical thinking abilities are in the decision loop.

We are training young people with exactly those skills. Teaching cryptography is an area where bringing more critical thinking into the classroom is absolutely vital. In fact, everyone around the world who deals with computer security and cryptography needs to evaluate claims carefully. In a very real sense, the security of modern civilization depends on skepticism and critical thinking.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the assistance of Mr. Mike Collins, a National Security Agency mathematician and a visiting professor of computer science at USAFA. We also note the contributions of Bruce Schneier's essay in [Sc94c], the founding document connecting skepticism and cryptography. This work should be required reading for critical thinkers everywhere.

The views expressed in this academic research paper are those of the authors and do not necessarily reflect the official policy or position of the US government, the Department of Defense, or the United States Air Force Academy.

REFERENCES

[Ad97] Adams, S., The Dilbert Principle, Collins Publishing, ISBN 0887307876, © 1996.

[AMI06] "AMI-Partners reports small and medium businesses ready to spend over $11B in IT security", June 26,2006, available at http://www.ami-partners.com/ami/sections/Press/Global_SMB_IT_Security_with_slide.pdf

[CC05], Common Criteria for Information Technology Security Evaluation, http://www.niap-ccevs.org/assets/images/cc_v23_part1.gif.

[Dy79] Dyson, F. "Time Without End: Physics and Biology in an Open Universe", *Reviews of Modern Physics*, Vol 52 No 3, July 1979, pp 447-460.

[Go06], Gorman S., "Hacker attacks hitting Pentagon", Baltimore Sun, July 2, 2006

[Kr06], Krebs B., "OMB sets guidelines for federal employee laptop security", Washington Post, July 2, 2006

[NAS96] Cryptography's Role in Securing the Information Society, report from the National Academy of Sciences/National Research Council, 1996, Kenneth W. Dam and Herbert S. Lin, Editors, available at http://newton.nap.edu/html/crisis/

[NIAP05] National Information Assurance Partnership, "Common Criteria for Information Technology Security Evaluation", available at http://niap.bahialab.com/cc-scheme/cc_docs/cc_v23_part1.pdf

[NiCh00] Nielsen, M. and Chuang, I., Quantum Computing and Quantum Information, Cambridge University Press, ISBN 0521635039, © 2000.

[Ra06] Radicati Group LTD, Email Security Market, available at http://www.radicati.com/brochure.asp?id=258'

[RSA05] RSA Security Inc. Corporate Press Kit, available at http://www.rsasecurity.com/node.asp?id=1382

[Sc94a] Schneier, B., Heartland Perspectives, The Heartland Institute, June 23, 1994.

[Sc94b] Schneier, B. Applied Cryptography, John Wiley & Sons, ISBN 0471117099, 2nd Edition, © 1994.

[Sc94c] Schneier, B. “Snake Oil”, Crypto-Gram Newsletter, February 15th 1999 , http://www.schneier.com/crypto-gram-9902.html#news

[Yu96] Yu, H. et al, “Teaching a Web Security Course to Practice Information Assurance”, Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education, March 2006, Houston TX.

[Zi97] Zimmerman, P. “Beware of Snake Oil“, 1991, excerpted from PGP User’s Guide, updated 1997, http://www.philzimmermann.com/EN/essays/SnakeOil.html.

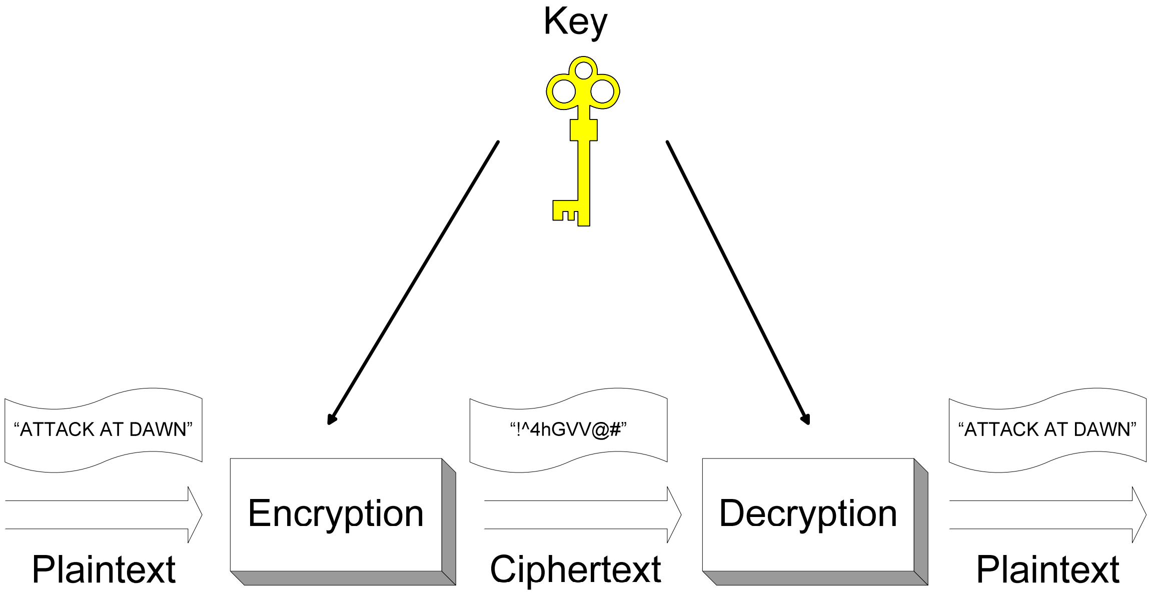

Figure 1: A Symmetric Cryptosystem

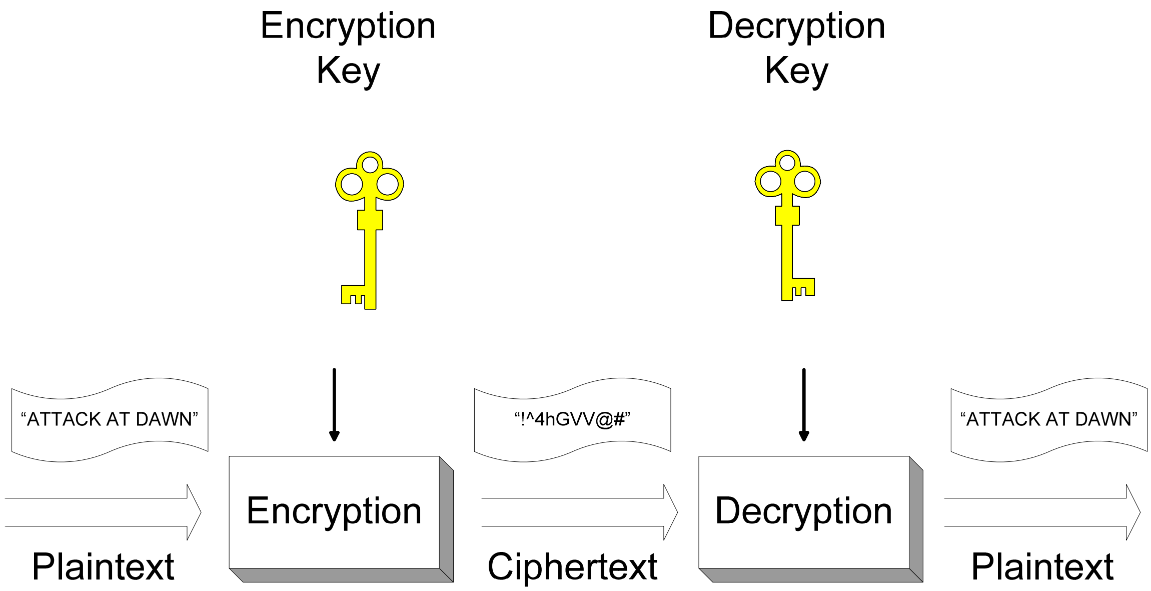

Figure 2: An Asymmetric Cryptosystem

Plain	Cipher
A	D
B	H
C	Q
D	L
E	A
F	Z
G	X
H	K
I	O
J	C
K	V
L	G
M	F
N	J
O	Y
P	R
Q	U
R	W
S	S
T	T
U	I
V	B
W	N
X	M
Y	E
Z	P

Table 1: Sample Key for Substitution Cipher