Insert That Odd Thing I Can't Ever Come Up with Here

Submitted in fulfillment of English III requirement
for the

Bachelor of Science
in
Mathematics

Submitted by

Kristi Short

Under the guidance of
Dr. Steven Butler

Department of Mathematics
Iowa State University of Science and Technology
Ames, Iowa – 50014

Fall Semester 2015

Contents

1 Introduction **2**

2 Counterexamples Win Out **3**

3 Introductory Cryptographic Terminology and Notation **4**
3.1 Asymmetric & Symmetric Cryptography 4
3.2 Connecting Symmetric & Asymmetric Cryptography 6

4 Semantic Security: Defining Security with Games **7**
4.1 Goldwasser & Micali's: Mental Poker and Partial Information 7
4.2 Weaknesses in the Assumptions of RSA 7

5 Attack Models **9**
5.1 Known Plaintext Attack (KPA) 9
5.2 Chosen Plaintext Attacks (CPA) 9
5.3 Chosen Ciphertext Attacks (CCA) 10

6 A Brief History of Lattice-based Cryptography **12**

7 Learning with Errors (LWE) **13**
7.1 Learning with Parity (LWP) 13
7.2 Noisy Learning with Parity 13

8 Soliloquy Construction **14**

1 Introduction

Modern cryptographic schemes can generally be divided into two classes: asymmetric-key schemes and symmetric-key schemes.
Symmetric-key schemes involve only one secret key between two communicating parties called a *shared secret*. In contrast, asymmetric-key schemes contain two secret keys, one for each party, and two public keys which are publicly distributed. Symmetric schemes are in general more efficient than asymetric schemes but have the disadvantage that keys must be agreed upon by parties who wish to communicate in advance. Specifically symmetric-key schemes require that secret keys be distributed over secure channels [1]. These types of key distribution schemes are not compatible with most modern methods of communication over networks, which are by and large conducted over the internet, an insecure channel.
In contrast, Public-key cryptography is an asymmetric-key class of systems which allows parties which have never met, much less agreed upon secret keys in advance, to communicate privately over insecure channels. The primary concern is to make communication by public key cryptographic schemes as efficient, secure, and easy to implement as those of symmetric key systems [2].
In this context, implementation of a particular cryptographic scheme is often a two-fold problem. That is, on one hand the cryptosystem must be constructed with much thought in mind to its efficiency and economy, on the other hand improper assumptions concerning the security imparted by the system must always be given precedence as any failure in this regard invalidates the inherent purpose of the system's existence, making all other efforts with respect to the system trivial in it's wake. Since the purpose of cryptography is to keep communications and information private the role in managing the risks associated with the system must be weighted necessarily heavier towards ensuring the fulfillment of a given set of security objectives. A balance between efficiency and security must be carefully, and continuously, maintained throughout construction, implementation, and daily maintenance of the system. To even momentarily neglect this precarious balancing act courts disaster.

[1] In real world applications there is no such thing as a secure channel.

[2] Notably, the problem of determining if any message has indeed been communicated securely (or if it has not) is an open problem for which no known solutions exist that cannot be classified as temporary hacks.

2 Counterexamples Win Out

This story begins at one such occurance of forgetfulness in modern cryptography. Where the balancing act between efficiency and security in lattice-based cryptography was for a time forgotten, and constructions by many were weighted towards efficiency. Particularly in ideal lattice-based cryptography, where a great many researchers took for granted the level of security lattice-based schemes were conjectured to impart, focusing their attention on the efficiency of their schemes instead. These researchers attempted construction of more and more efficient schemes, assuming that the possiblity their systems were as secure as schemes defined over more general lattices, would eventually be proven,... by someone else.

In mathematics the (and possibly life in general), the best possible refutation of a claim something does not or cannot exist is a counterexample. Such an example stands constructed in all its defining premises, defiantly and conclusively proving *"I exist".* Cryptography is not immune from these somewhat ammusing and ironic examples, and lattice cryptography is no exception to this rule, where these counterexamples often show themselves in the form of use cases.

The faith of the ideal lattice cryptographers in the strength of their efficient constructions is not a new mistake in the history of cryptography. This assumption of strength is famously mirrored in the arogance of engineers of the German Enigma ciphers of World War II, stubbornly clinging to the calculations of theoretical strength despite all evidence to the contrary. Doubtless, history professors with interest in such affairs would be fully justified in gloating and repetition of the adage *"those who forget history are doomed to repeat it,"* with all the smuggness they can bear.

There is no better such example in modern lattice-based cryptography then the Soliloquy problem; an unexpected announcement by British researchers in the Communications-Electronics Security Group (CESG) (the information assurance division of the Government Communications Headquarters (GCHQ)). This informal report dubbed *Soliloquy: A Cautionary Tale* [[**Cam20140**]], brought on a slew of uncharacteristically emotional responses in the cryptographic community, a mixture of shock, speculation, as well as a few cases of well-earned, smug *"I told you so"* sentiments. As for researchers working towards efficient constructions of similar schemes there was a backlash of reactions ranging the spectrum of in one direction, unsurprised acceptance and in the opposite direction, rage and denial.

The announcement by CESG, vaguely outlined a cryptosystem given over ideal lattices of characteristic two, a characteristic much overused despite the warnings of more experienced researchers [[**Cra20151**]] that efforts in other fields were necessary in order to guard against the possibility that the particular field characteristic was shown to exhibit such a vulnerability.

3 Introductory Cryptographic Terminology and Notation

Public Key Cryptography (PKC) solves the problem of how to enable two people who have never met to communicate securely over insecure channels.
Public-key depends on the following assumptions:

1. In the forward direction, we assume there exists a one-way function which is easy to compute.

2. In the reverse direction, we assume that the same function is hard to invert.

INSERT PKC DIAGRAM

3.1 Asymmetric & Symmetric Cryptography

The concepts of symmetric and asymmetric cryptographic-key schemes are given below as independent forms. However, it is rarely the case that these schemes will be applied in this manner particularly in a Public-key asymmetric system. Both asymmetric and symmetric key schemes are assumed to be classical cryptosystems. By way of classical we mean the systems are defined in terms of the chances the mathematical problems these systems are based upon have of being solved by an adversarial system. I.e., The liklihood of breaking the security of the ciphers in any of these systems is dependent on the assumed capabilities of the adversary's system. The common assumptions attributed to an adversary are as follows:

- The adversary is in possession of (or has access to) slightly more than a reasonable (but still finite and non-quantum) amount of computational resources;

- If the adversarial system (as described above) provides a solution to the problem serving as a basis for security of the target system, it does so by applying a probabilistic, polynomial-time, algorithm.

Symmetric-key Cryptography

In cannonical symmetric-key schemes only one key is generated for both the encryption and decryption algorithms. Symmetric-keys must therefore be exchanged over secure channels where both communicating parties have agreed in advance to the method of encryption and have exchanged secret keys.
Formally, we say the two parties are in possesion of a *shared secret* and each party is mutually, as well as equally responsible for the maintenance of the secret which establishes and secures their communications.
In certain contexts this method can be made to be as secure as an asymmetric key system. It's main drawback is the shared secret which has the ability to invalidate the integrity of communications if either party is compromised.

Is this supposed to be here, I think this sentence is supposed

Additionally, since the method requires a secure channel as well as a shared secret both parties must know they will have a need for encrypted communication in advance and negotiate the means of transmission and key exchange prior to communicating. This means the parties must know each other prior to communicating.

Formal Definitions for Symmetric-key Schemes

We define a symmetric-key scheme as follows:
Let $\mathcal{K}$ be the set of all possible keys k generated by a key generation function, called the *keyspace*.
The key generation algorithm **Gen** is a probabilistic function which uses a distribution chosen which is appropriate with respect to the scheme.
The key generation algorithm randomly chooses a key k from it's distribution uniformly.
Let $\mathcal{M}$ be the set of all possible messages m called the *message* or *plaintext space*.
The encryption function **Enc** takes a key k and message m as input producing a ciphertext c.
The set of all possible ciphertexts (The ciphertext space) is denoted $\mathcal{C}$, defined by taking the pair of spaces $(\mathcal{K}, \mathcal{M})$ together.
Let **Dec** be the decryption function which takes a ciphertext c and key k as input returning the corresponding message m.

Cryptosystem

Taking the above components together we have the following defintion of a symmetric-key cryptosystem using the traditionally chosen communicating parties *Alice, Bob* and *Eve (the eavesdropper)*:
A symmetric-key cryptosystem is given by a set of algorthms $(\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec})$ and the message space $\mathcal{M}$.
The cryptosystem performs it's functions in the following order:
First, Alice and Bob use *Gen* to produce a shared secret key k.
Whenever Bob wishes to send a message m to Alice (or Alice to Bob) he inputs (m, k) into *Enc* to get a ciphertext of the message c.

$$c \coloneqq Enc_k(m)$$

When Alice recives the ciphertext c from Bob she inputs both (c, k) into the decryption function and gets back the plaintext message m.

$$Dec_k(c) \coloneqq m.$$

To ensure the cryptosystem is correct, that is that it holds *for every* message key pair $(k, m) : k \in \mathcal{K}$ and $m \in \mathcal{M}$ you must show that using the key on a ciphertext outputs the correct message.

$$Dec_k(Enc_k(m)) = m$$

$$Dec_k(c) = m : Enc_k(m) = c$$

Asymmetric-key Cryptography

In contrast symmetric-key schemes are determined by two sets of keys one for each party. Each set consists of a private key known only to the owner and a *public key* which can be widely distributed. Public-key uses a *one-way function* as its public key; this function is easy to compute but hard to invert and is available publicly for use as an encryption method for the key owner. A second key called the private key is known only to the key owner; this key is a *trapdoor function* which can invert a ciphertext back to its plaintext form.

Insert equations as above

3.2 Connecting Symmetric & Asymmetric Cryptography

It seems at first glance that asymmetric and symmetric cryptosystems are isolated methods of encryption. However, symmetric encryption is well-suited for the task of assisting asymmetric key schemes not only for efficiency purposes but also in the role of key management.

4 Semantic Security: Defining Security with Games

4.1 Goldwasser & Micali's: Mental Poker and Partial Information

Semantic Security was a game proposed by Goldwasser and Micali in their 1982 paper entitled: *"Probabilistic Encryption & How to Play Mental Poker Keeping Secret All Partial Information"*[[**Gol19820**]], in response to papers from Rivest, Shamir, Adelman, [[**Riv19780**]] and Rabin [[**Rab19790**]] detailing the theory behind the Diffie-Hellman based RSA cryptosystem.
Goldwasser and Micali's intention was to strengthen the security assumptions of Diffie-Hellman based cryptosystems by formally defining what it would mean for such systems to be called secure.
[[**Gol19820**]] Proposed the following property for any implementation of a Diffie-Hellman Public-key Cryptosystem:

Proposition 1. *"An adversary, who knows the encryption of an algorithm and is given the cyphertext, cannot obtain any information about the cleartext."* [3]

Informally, Goldwasser and Micali's property says that a given cryptographic scheme is considered insecure if given some of the ciphertext (but not the private key), it is possible for an adversary to recover any information about the plaintext, or any recover useful information about the plaintext of the message, by some manipulation of the ciphertext over a reasonable amount of time.

4.2 Weaknesses in the Assumptions of RSA

Goldwasser and Micali pointed out that the security assumptions given in [[**Rab19790**]] and [[**Riv19780**]] had some particularly significant weaknesses that shouldn't be overlooked.

Assumption 1. There exists a trapdoor function $f(x)$ that is easily computed, while x is not easily computed from $f(x)$ unless some additional information is known. To encrypt any message m, you evaluate f at m, and receive the ciphertext $f(m)$.

[[**Gol19820**]] give two weaknesses of this first assumption:

Remark 1. If x has some specialized form, the fact that $f(x)$ is assumed to be a trapdoor function does not guarantee that x cannot be computed. There are two important points in this weakness:

Case 1. First, messages do not typically consists of randomly chosen numbers, they have more structure.

[3] [**Gol1982**] Pg.1, Paragraph 1, Lines 3-6

Case 2. Second, the additional structure these messages posess may assist in decoding the ciphertext.

Example 1. Compare two functions $f(m)$ and $g(n)$, where the input m for f is some random ASCII sequence, and the input n for g is an ASCII sequence which represents sentences written in English. Then it may happen that m is hard to recover from $f(m)$, but that n is easy to recover from $g(n)$.

Remark 2. Even if we assume that f is a trapdoor function, that does not negate the possibility that the message or even half the message may not be recoverable from $f(x)$.

The second weakness of RSA concerns the trapdoor function itself:

5 Attack Models

5.1 Known Plaintext Attack (KPA)

The adversary in this case has at least one pair of corresponding plaintext and ciphertext that they did not choose. The adversary knows the context or meaning of the plaintext. The adversary does not have access to the system and cannot generate more pairs of plain and ciphertext; they must work only with what they have.

- The adversary knows the system.
- The adversary has one ciphertext and a corresponding ciphertext.
- **Goal:** The adversarys job is then to determine the decryption key.

Example
Eve has heard a conversation between Alice and Bob and knows that they will meet each other 'next Monday at 2pm'. This particular conversation that Eve overheard is encrypted and Eve has the ciphertext and she knows what was said so she has the plaintext. So Eve knows somewhere in the ciphertext is some string corresponding to 'next Monday at 2pm'. Eve can then use that she knows this string to decode some portion of the ciphertext and from there try to deduce the secret key.
Specific Examples

- Cesar Ciphers
- The Polish Break of the Enigma Machine
- The Bletchley Park Break of the Enigma Machine
- Old versions of PKZIP encrypted with AES

5.2 Chosen Plaintext Attacks (CPA)

Chosen Plaintext Attack I (Batch Chosen Plaintext Attack)

- The adversary knows the system.
- The adversary has temporary access to the encryption mechanism containing the key.
- **Goal:** For the time the adversary has access to the mechanism they must choose plaintexts, construct the corresponding ciphertexts and deduce the decryption key somewhere in this or after this process.

Chosen Plaintext Attack II (Adaptive Chosen Plaintext Attack)

- The adversary knows the system.
- The adversary has unlimited access to the encryption mechanism containing the key.
- **Goal:** The adversary can then choose plaintexts, see what the system returns, intelligently decide upon and submit more plaintexts to the system and use the constructed corresponding ciphertexts to deduce the secret key (trial and error).

Example
A very badly designed email service for some company provides encryption of messages, however it only generates one key per branch. Eve knows that Alice and Bob work at a particular branch of a company gets a job at the company transfers to that branch. Eve uses the same email service sends email with content she chooses through the service (perhaps to herself BCCing or CCing herself it doesnt matter, Eve chooses the content), Eve then analyzes the ciphertexts of the emails she sent deduces the secret key and then proceeds to decrypt messages belonging to Alice and Bob.

5.3 Chosen Ciphertext Attacks (CCA)

Chosen Ciphertext Attack I (Lunchtime Attack)

- The adversary knows the system.
- The adversary has temporary access to the decryption mechanism containing the key.
- The adversary has one or more ciphertexts which correspond to this key.
- **Goal:** The job of the adversary is generate the plaintext corresponding to the ciphertexts (easily done), and deduce the key.

Example
Eve has acquired some encrypted messages sent from Bob to Alice. She has gained access to Bobs computer (the decryption oracle) while Bob is at lunch. Eve gets access to Bobs computer while he is at lunch and decrypts the set of encrypted messages. Eve then uses the ciphertexts and plaintexts to deduce the secret key.

Chosen Ciphertext Attack II (Adaptive Chosen Ciphertexts Attack)

- The adversary knows the system.
- The adversary has unlimited access to the decryption mechanism containing the key.
- The adversary has one or more ciphertexts which correspond to this key, and can now generate more whenever they choose.

- **Goal:** The job of the adversary is to generate the plaintext corresponding to the ciphertexts (easily done), use the information to intelligently choose more plaintexts to generate ciphertexts and in this process deduce the key.

6 A Brief History of Lattice-based Cryptography

Rework this section altogether

Lattices (Informally)

This subsection gives a brief history of lattices before their modern uses.

Before lattices became popular in cryptography circles they were in terms of mathematical history by and large mocked, ridiculed, maybe even hated. Lattices can be traced back to Gauss' circle problem (a favorite IMO strategy for solving problems in synthetic geometry), which asks *given a circle* $\mathcal{R}^2$ *centered at* $\mathcal{O}$ *how many integer lattice points* (m, n) *are inside* $\mathcal{R}$*?*

In 2009, lattice cryptography began a swift resurgance in popularity after Oded Regev's new problem defined over lattices called *Learning with Errors* (LWE) [[**reg2009**]] proved to be as hard to solve as certain worst-case lattice problems. Learning with Errors is a generalization of the *Parity Learning with Noise* (PLN) problem which is conjectured to be hard to solve. [4].

[4]Both PLN and LWE are described in section

7 Learning with Errors (LWE)

Learning with Errors is a lattice problem based upon the machine learning problem called Noisy Learning with Parity (NLWP).

7.1 Learning with Parity (LWP)

An algorithm trained to solve the Learning with Parity problem is given the following parameters:

- Samples $(x, f(x))$, where the samples are generated by some distribution over the input.
- For the function f the algorithm has the *assurance* that f can compute the parity of bits at one or more fixed locations.

If the algorithm solves the LWP problem it has sucessfully guessed the original function f from the given samples.
The above version of LWP is easy to solve using Gaussian elimination provided the algorithm is given enough samples, and the distribution from which the samples are drawn is not overly skewed.

7.2 Noisy Learning with Parity

Noisy Learning with Parity (NLWP) deviates slightly from the constraints given above significantly alterning the solvability of the problem.
The algorithm's parameters are adjusted as follows:

- The algorithm is provided with samples (x, y), such that $y = 1 - f(x)$ has only a small probability of knowing the parity of bits at some locations.
- Additionally, these samples may contain errors (noise).

The adjustments to the LWP problem change the expected outcome from easy to solve to that of one conjectured to be hard to solve.

8 Soliloquy Construction

Let $n \in \mathbb{P} : |n| \approx 10$ bits. Let ζ be a primitive n^{th} root of unity.
We define K to be the number field $\mathbb{Q}(\zeta)$ and (O) the ring of integers $\mathbb{Z}[\zeta] \in K$.
Select α as a candidate for $p_k = \sum_{i=1}^{n}(\alpha_i \zeta_i) \in \mathcal{O}$, such that the coefficients α_i are taken from a discrete Gaussian distribution with a mean of zero. i.e. The coefficients are relatively small.
Let p be the norm of α.
The conditions under which an ordered pair, (α, p) can be used as a set of Soliloquy keys (pk, sk), are

- $p \in \mathbb{P}$ and
- $c = 2^{\frac{p-1}{n}} \not\equiv 1(modp)$.
- Then the quantity c occurs with probability $1 - \frac{1}{n}$.

Using these conditions we guarantee that we have at least one principal ideal of the form $\mathcal{P} = p\mathcal{O} + (\zeta - c)\mathcal{O}$, where a prime $p \equiv 1(modn)$ is the principal ideal p_O for a particularly chosen $c = 2^{\frac{(p-1)}{n}}$.
If p_O is a product of prime ideals $\mathcal{P}_i$ with representation $\mathcal{P}_i = p\mathcal{O} + (\zeta - c_i)\mathcal{O}$ where the c_i are the non-trivial n_{th} of unity mod p.
Since the Galois group $Gal(K/\mathbb{Q})$ gives a permutation of prime ideals $\mathcal{P}_i$, then by a reordering of the coefficients a_i, (i.e. taking some Galois conjugate of α), we can ensure $\alpha\mathcal{O} = \mathcal{P}$.
If $p\mathcal{O}$ is a product of prime ideals $\mathcal{P}_i$ with representation:

$$\mathcal{P}_i = p\mathcal{O} + (\zeta - c_i)\mathcal{O},$$

where the c_i are the non-trivial n^{th} roots of unity mod p.
Since the Galois group $Gal(K/\mathbb{Q})$, gives a permutation of prime ideals $\mathcal{P}_i$, by a reordering of the coefficients a_i,
(*i.e. taking some Galois conjugate of* α), we can ensure $\alpha\mathcal{O} = \mathcal{P}$.
We have obtained a natural homomorphism

$$\psi : \mathcal{O} \to \mathcal{O}/\mathcal{P} \simeq \mathbb{F}_p$$

such that:

$$\psi(\epsilon) = \psi\left(\sum_{i=1}^{n} e_i c^i\right) = \sum_{i=0}^{n} e_i c^i \mod p = z.$$

Appendix A: Gauss' Circle Problem

Given a circle $\mathcal{R}^2$ with radius $r \geq 0$ centered at the origin $\mathcal{O}$, how many integer lattice points (m, n) can you find in the circle?
Since we can express a circle in terms of Euclidean coordinates as:

$$x^2 + y^2 = r^2.$$

We can state the question in terms of our lattice points (m, n) as:

$$m^2 + n^2 \leq r^2.$$

Approximate Solution

The area of the inside of a circle is approximately $A \approx \pi r^2$.
Which gives us approximate solution in terms of r for Gauss' circle problem (denoted $N(r)$) of:

$$N(r) = \pi r^2 + E(r),$$

where $E(r)$ is some error term given by the radius.

Exact Solutions

Two forms for an exact solution for the circle problem can be given in terms of sums.
The first is a bit ugly:

$$N(r) = 1 + 4\sum_{i=0}^{\infty} \left(\lfloor \frac{r^2}{4i+1} \rfloor - \lfloor \frac{r^2}{4i+3} \rfloor \right).$$

The second solution has a much nicer form if the sum of squares is defined as the number of ways of writing n as the sum of two squares:

$$N(r) = \sum_{n=0}^{r^2} r_2(n).$$

Todo list

INSERT PKC DIAGRAM 4
Is this supposed to be here, I think this sentence is supposed to be in asym. . . . 4
Insert equations as above 6
Rework this section altogether 12
This subsection gives a brief history of lattices before their modern uses. 12
Insert Section Number Here 12
thebibliography
gls