# An Exploration of Modern Cryptography

Siddharth Mahendraker
siddharth_mahen@me.com

International School of Helsinki
Supervisors: Seth Tyler & Jyri-Pekka Komonen
Word Count: 3246

2012

**Acknowledgements**

I am most sincerely grateful to my personal project supervisors, Mr. Tyler and Mr. Komonen, whose guidance and feedback was invaluable in the completion of this project.

# Contents

# The Goal

The realm of computer science has captivated me since my youth. It has always been an interest of mine which I have pursued inside and outside of school. These days, I continue to write programs in my spare time, publish open source software, and contribute to other open source projects over the Internet.

Therefore, when the time came to pick a topic for my personal project, a topic in the field of computer science seemed like an obvious choice.

Through my own interest, I had already taken time to learn about many of the interesting subtopics in computer science, such as computational complexity theory, artificial intelligence, algorithm and data structure design and software engineering. As such, I wanted to try something a little different; something new and challenging.

I chose to learn about cryptography because it was a field of computer science I genuinely knew nothing about, yet contained so many interesting ideas. What really drew me in was the fundamental problem that cryptography attempts to solve: it allows people to communicate securely over completely insecure channels. I was really interested in learning about the mathematical theory underneath cryptography and how it could be implemented in the real world.

I think the area of interaction to which cryptography most relates is human ingenuity. This is because cryptography (and for that matter any field in computer science) is based on using abstract concepts to find creative solutions to concrete problems. In cryptography, these abstract concepts take the form mathematical equations, these solutions take the form of cryptographic algorithms and the problems involve communicating securely.

Here is a brief example of a simple problem with an ingenious cryptographic solution. Suppose two people, Alice and Bob want to know a shared secret code, but all of their communications have to be public. The ingenious solution to this problem, known as the Diffie-Hellman key exchange algorithm, works as follows. Alice and Bob both agree on a public prime number, and a generator, $p$ and $g$ respectively. Then both Alice and Bob choose random numbers, say $a$ and $b$, and compute $A = g^a \pmod{p}$ and $B = g^b \pmod{p}$ respectively, which are then sent to one another. Now Alice, having received $B$ from Bob, computes

$$B^a = (g^b)^a = g^{ab} \pmod{p}$$

And Bob, having received $A$ from Alice, computes

$$A^b = (g^a)^b = g^{ab} \pmod{p}$$

Now they both share the same secret number $g^{ab} \pmod{p}$ and all everyone else has seen are the values $A$, $B$, $p$ and $g$!

Based on my area of interaction and my interests, human ingenuity and learning about both applied and theoretical cryptography respectively, I came up with the following inquiry question: "What ingenious ideas and processes are involved in modern cryptographic theory, and how do they function in practise?".

I decided that the best way to achieve my goal and answer my inquiry question was to analyze and compare three separate cryptographic algorithms embodying the fundamental concepts of cryptography and compile this work as a research report. This goal is particularly well suited to my needs as it balances both the applied and theoretical aspects of cryptography, allowing me to focus on the entire subject as a whole.

My product should follow the following specifications:

1. Firstly, it should cover the three building blocks of cryptography: stream ciphers, block ciphers and public-key ciphers. In this regard, it should be complete and the ideas should flow clearly. Everything that occurs should be explained, and there should not be any "magic" going on between paragraphs. This means there should be many worked examples throughout the text, and at least one for each key concept. This criterion ensures my final product covers all of the (elementary) processes involved in modern cryptography, as stated in my inquiry question. I also think my product should be between 20 to 30 pages long.

2. Secondly, my product should include a mix of theory and real life applications. As my inquiry question states, this is as much about practical implementation as it is theory. This also means that my product should provide code examples of all of the algorithms presented, as this directly relates to their application in real life. Furthermore, the reader can then experiment and check the validity of my work, and in the process become even more familiar with cryptography.

3. Thirdly, it should look good and be entertaining to read. If I am going to write about a topic that I spent a whole year researching, the paper I submit should convey some of my passion for the subject. I

think that the human ingenuity behind cryptography is absolutely amazing, and the reader should feel the same way once they have read my report.

## Selection of Sources

My research spanned a wide variety of subjects in both theoretical and practical cryptography.

One of the most useful books I have used is Applied Cryptography by Bruce Schneier [8]. It gave me invaluable information into the practical application of cryptography with solid definitions, great breadth and full code examples. I was sure this book was a good resource because the author, Bruce Schneier, is a very well known security technologist, who currently serves on the board of directors of the International Association of Cryptologic Research and is a member of the Advisory Board for the Electronic Privacy Information Center while also working at a consulting firm specializing in cryptography and computer security. Furthermore, he has also designed one of the strongest block ciphers algorithms currently available in the public domain, called Blowfish [7]. I found this book by asking about good introductory texts to cryptography on an Internet forum I frequent, called Quora [6].

The next invaluable resource I used was An Introduction to Mathematical Cryptography by Joseph H. Silverman et al [3]. This book gave me a complete and through understanding of the underlying mathematics behind a wide variety of cryptographic algorithms, as well as insight into interesting mathematical techniques which could be used to break these algorithms. The book was initially an extremely tough read, as I was not used to its rigorous proof based method. However, with practise and much time spent on the wonderful examples at the end of each chapter, I was able to use this book effectively. I believe this book contains trustworthy information for several reasons. Firstly, all of the authors of this book are professors of mathematics at Brown university, which is well known for its strong and academically active research group in number theory and algebraic geometry, both of which are highly relevant to cryptography. Specifically, Silverman is a very well known number theorist who has published numerous publications in these fields, including papers such as The Arithmetic of Elliptic Curves and A Friendly Introduction to Number Theory.

Another huge resource which I took advantage of was websites such as Khan Academy and Wikibooks [4, 9]. Khan Academy creates and distributes short Youtube videos explaining topics in a wide variety of disciplines. Wikibooks provides free, detailed texts regarding a wide variety of subjects. Although these resources were not used when looking for information regarding topics in cryptography, they were extremely useful when I wanted to review certain topics, such as binomial coefficients or learn how to use the typesetting software (LaTeX), properly. Although these may not have been the most genuine sources as neither Wikibooks contributors nor Salman Kahn publish peer reviewed material, however, they certainly did give me the information I needed quickly and succinctly.

Aside from these resources, I have also referred to a many different individual articles from various online sources, ranging from an article on elliptic curve cryptography from the National Security Agency (NSA) [1] to articles from Stephen Wolfram regarding the applications of cellular automata in cryptography [10]. The NSA is a very well recognized agency, and is one of the global leaders in cryptographic research. They have contributed some of the most important algorithms to date, including DES (Data Encryption Standard), and the cryptographic hashing algorithms SHA-1 and SHA-2. For these reasons, I can be sure that the data they provide is accurate. Likewise, I know Stephen Wolfram's research is also trustworthy as he is the inventor of the Mathematica software package for computation and has previously published very important works regarding cellular automata and complexity theory.

When it came to picking my resources, I opted for quality over quantity. Although I did not choose an enormous selection of resources as some students have, the resources I did choose were of excellent quality and were ultra dense in information. For example, both of the books mentioned above are well over 450 pages in length (501 and 758 pages respectively).

# Application of Information

Over the course of my project, there were a slew of skills and techniques I developed as a result of my research, which concentrated themselves in two main categories.

Firstly, and perhaps most importantly, I learnt about the actual processes involved in the use and the construction of cryptographic systems. I learnt

about techniques such as statistical analysis of text, linear cryptanalysis and the analysis of elliptic curve discrete logarithm based cryptosystems, just to name a few. These techniques were applied directly to my product in two distinct ways. Firstly, all of these ideas were explained in detail, with worked examples. This should demonstrate my ability to apply this information through my ability to teach it to others. Secondly, I have written programs for both the encryption and the cracking of each of the algorithms in my product. This should further demonstrate my understanding of the subject.

Secondly, I learnt to apply many auxiliary skills to achieve my project goals. These skills were not central to answering my inquiry question, but rather served to enhance and improve my project as a whole. One of the most important example of this was my development of my ability to read and understand mathematical texts. At the beginning of the project, I had absolutely no idea how to read mathematics, which resulted in me flipping back between pages trying to understand what was happening during a proof or when building on a previous topic. As the project progressed, I learnt to read carefully and use my resources properly (for example, by doing the questions at the end of each chapter, or working all of the examples by hand). Furthermore, apart from helping me understand the texts I was reading, this helped me structure my writing style appropriately, so I sound clear and professional throughout my research report. Another great example of the skills I learnt to apply are my new found LaTeX typesetting skills. Before this project began, I had not even heard of LaTeX, but after doing some research on better typesetting engines (Microsoft Word does not set mathematical text very well...) I chanced upon LaTeX and took it upon myself to learn how to use it. The wonderful Wikibook on LaTeX provided me with everything I needed to know. Using this knowledge, I constructed not only my product, but also this essay.

I realize this may sound haughty, but I truly feel that anyone who is to grade me on my application of information should really read my product to truly understand the extent to which I have applied the information I learnt. This being a reflective essay, the technical content of my project has only been vaguely mentioned (as above). However, in reality, my depth and application of knowledge is quite quite substantial and I think this should be taken into consideration. Here is a brief example of my application of such technical information: Early on in the project, I was having trouble understanding why Shank's baby step giant step collision algorithm for the discrete logarithm problem [3, p. 63], $g^x = h$, required lists of size slightly larger than the size of the square root of the order of the

element $g$. It seemed absolutely strange at first, however, as I continued through the book, I learnt about the Collision Theorem [3, p. 228] and how it (practically) guarantees that you only need to check a small multiple of the size of the square root of the list before you find a match, because by that point the probability of not finding a match is very very low.

## Achieving the Goal

My goal was to answer the question "What ingenious ideas and processes are involved in modern cryptographic theory, and how do they function in practise?", and I think that I have succeeded in achieving that goal with regard to my product.

My research report meets all of the specifications I set out to achieve:

1. I have covered the three principle types of cryptographic algorithms present in modern cryptography: stream cipher, block ciphers and public-key ciphers. The text is 28 pages long, not including the appendix. This answers the "what ingenious ideas and processes are involved in modern cryptographic theory" and meets my page count specification.
2. I have made sure that the text flows clearly and that there are many examples throughout the text. Specifically, I have included one fully worked example for each key concept. This also addresses the "what ingenious ideas and processes are involved in modern cryptographic theory" part of my inquiry question.
3. I have reflected and evaluated each cipher with regard to its theoretical and practical advantages/limitations. Furthermore, I have provided full commented code examples of all of the cryptography algorithms analyzed. This addresses the "how do they function in practise" part of my inquiry question.
4. I have made sure (to the best of my ability) that my product is good looking and entertaining to read. It has been set using LaTeX, and includes gorgeous tables and beautiful mathematical typesetting. The paper flows quite nicely, and although far from being professionally written, conveys my passion and interest in the subject.

Based on all of the above, I would give myself a level 4. I believe I have created a product of exceptionally high quality which has met all of the

specifications I set out to achieve. I think this is a just score because my criteria are sufficiently concrete and rigorous (you can read my report and check each one off) and each one relates directly to my inquiry question (except point 4, which was a purely stylistic goal).

## Reflecting on Learning

The completion of my project as has extended my knowledge and understanding of cryptography, human ingenuity and myself on a very profound level.

During my project, I learnt about a myriad of topics ranging from the construction and analysis of cryptographic algorithms to number theory to elliptic curve arithmetic. One of the most important parts of this learning was the reading and understanding of complex mathematical texts. At the beginning of my project, I was struggling to read mathematical texts because I would read them like a novel. Soon, I realized that truly understanding these texts involves reading, re-reading, analyzing and working through examples. When my first theorem finally "clicked" I was overjoyed. The beauty of simplicity of the idea were all there for me to behold, and I was ecstatic to finally understand its statement on a deeper more primordial level. When I finally understood not only how the proof was derived, but what it implied and how those implications could be used, I was given a glimpse into the minds of these text's authors, and the ingenious way in which they solved problems. This not only contributed significantly to the clarity of my product, but also to my personal development in proof construction, logical deduction and scientific writing style.

As this project project progressed, I came to a very profound conclusion regarding cryptography, mathematics and the nature of human ingenuity. I realized that simply by making up structure, and imagining new possibilities, humans have the capacity to reveal deep truths about our universe. And in that respect, human achievement is only limited by out imaginations. Although this idea seems trivially simple, it has changed the way I think about science, innovation and especially, mathematics. I have come to see the intrinsic beauty of mathematical structure, and its ability to reveal amazing and wonderful truths through the sheer power of imagination. A great example of these ideas can be found in the Diffie-Hellman algorithm I briefly went over earlier, which allows two people to compute a shared secret even when all of their communications are in

public! What a perfect example of ingenuity. The abstract concepts of multiplication over prime finite fields are used to solve this apparently impossible problem in an amazingly elegant fashion.

This project also taught me a lot about myself in relation to my learning. It pushed the limits of my perseverance and my patience. Sometimes, I would stare at my textbook for a good 45 minutes, trying to figure out why something worked the way it did. Other times, I would have to work 10 problems from the end of the chapter before I truly understood the implications of a theorem. I learnt to know when I was in the mood to understand something, and when I was wasting my time with a problem which required more thinking. I also realized the importance of occasional breaks, despite tight deadlines, which always helped me clear my mind and focus on the task at hand. This project also significantly improved my self direction. Before this project, I would complete work in a haphazard way, skipping from one task to the next and sometimes going off on long tangents which would eat all of my time. Nowadays, I create a todo list with my tasks in order of decreasing importance, which I attempt to complete in one continuous chunk of time (3 hours, for example), with light breaks in between.

Sadly, I only picked this skill up in the later months of my project. If I were to do a similar project in the future, I would improve my management of time by allocating more of it toward writing the research report and less of it toward doing research. Although thorough research was crucial to the success of my product, I felt as though I could progressed through my research report with less stress had I started working on it earlier. This would have also allowed me to have more time to edit and refine my work, which is always appreciated. I think I could achieve this next time by allocating work more granularly, rather than in large chunks. This way, I would be able to stop and asses myself more often, so that I can makes changes in my plan earlier in the process. For example, during this project, I allocated work in month based chunks, meaning all of my work was due at the end of the month. Rather than do this, I should have allocated work in week based chunks, so that my goals wouldn't seem as large and I would be able to make more amendments to my plan as the project progressed.

All in all, I am really glad I did this project. I feel as though I have really taken my abilities to their greatest potential. I learnt an incredible amount, not only about ingenuity in cryptography, but also about myself.

# Bibliography


[1] National Security Agency. The Case for Elliptic Curve Cryptography. http://1.usa.gov/IaIdvX, 2009.

[2] Howard M. Heys. A Tutorial on Linear and Differential Cryptanalysis. *Cryptologia*, XXVI(3):189–221, 2002.

[3] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. *An Introduction to Mathematical Cryptography*. Springer, 233 Spring Street, New York, N.Y., USA, 2008.

[4] Salman Khan. Probability. http://bit.ly/J9qxix, 2008. Specifically, videos on combinatorics.

[5] Matthew Musson. Attacking the Elliptic Curve Discrete Logarithm Problem. Master's thesis, Acadia University, 2006.

[6] Quora. What are some good introductory books on cryptography? http://b.qr.ae/ILngsi, 2011.

[7] Bruce Schneier. Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish). In *Fast Software Encryption, Cambridge Security Workshop Proceedings*, pages 191–204. Springer-Verlag, December 1993.

[8] Bruce Schneier. *Applied Cryptography*. John Wiley & Sons, Inc., 605 Third Avenue, New York, N.Y., USA, second edition, 1996.

[9] Wikibooks. Latex. http://bit.ly/L5YUN, 2011.

[10] Stephen Wolfram. Cryptography With Cellular Automata. http://bit.ly/IGZ8Zu, 1986.

# Appendix

## Journal Excerpts

**13/10/11**

### Information Theory

Information Theory defines the amount of information in a message as the minimum number of bits needed to encode all possible meanings of that message, assuming all meanings are equally likely.

The amount of information in a message 'M' is measured through the entropy of 'M', denoted 'H(M)'. In general, the entropy of a message is obtained by taking the log base 2 of the number of possible meanings, once again assuming each meaning is equally likely. This can be expressed as the equation:

H(M) = log2(n)

Such that 'n' is the number of possible meanings of the message.

Entropy also measures the uncertainty of the message, ergo, the number of plaintext bits needed to be recovered from the ciphertext for the plaintext to make sense.

This can be used to explain why it's more likely that a message to Bob starts with "Dear Bob" and not "}!Q$aqw&".

The rate of a language is the amount of information (in bits) that is stored in each letter of the language, given as 'r' by:

r = H(M)/N

Such that 'N' is the length of the message 'M'. The normal rate of English is around 1.3 bits/character. That is to say, the average English message has approx. 1.3 bits of information in each character.

The absolute rate of a language is the maximum number of bits that can be coded in each character. If there are 'L' characters in a language, it is said to have an absolute rate, 'R', of:

R = log2(L)

In English, this is around 4.7 bits/letter. However, this is not the actual rate of English, as English is highly redundant. The redundancy of English, 'D', is defined as:

D = R - r

The redundancy of English is 3.4 bits/letter, meaning that on average, English letters have 3.4 bits of redundant or superfluous information.

The measure of the entropy of a crypto system can be approximated by the function:

H(K) = log2(K)

Such that 'K' is the size of the key space (ergo size of the set of all possible key values). In general, the greater the entropy of a system, the harder it is to crack.

Cryptanalysis uses the natural redundancies of a language to reduce the number of possible plaintexts, as more than one expression can match the same meaning. The more redundant a language, the easier to analyse. This is why it is good practice to compress messages before encryption, to reduce the redundancy of the message, as well as the work required to encrypt/decrypt.

There are two basic techniques to obscure the redundancies in a plaintext message, confusion and diffusion.

Confusion obscures the relationship between the plaintext, the ciphertext and the key, making the search for redundancy and statistical patterns more difficult.

Diffusion dissipates the redundancy of the plaintext by spreading it out over the ciphertext, for example a transposition cipher like the columnar cipher simply rearranges the character of the plaintext. More advanced forms of diffusion can spread parts of the message through the entire message, rather than just transforming it.

Stream cipher use confusion only, block ciphers use both confusion and diffusion. As a rule of thumb, diffusion alone is always easily cracked.

**18/02/12**

## 18/02/12

Ski break is starting. My school work has finally subsided and I plan to complete my implementation of ECC. Finally understand all of the math behind it. This should not be that difficult.

ECC presents itself as a very interesting and efficient alternative to normal DLP-based ciphers. Check out these stats from the NSA. They state that to maintain a security level equivalent to that of a 3072 bit key in RSA (read strong DLP) you only need 256 bit key in ECDLP! Thats 10x less space cost! Furthermore, this decrease in space cost increases as the security bits are increased!

| Security Level(bits) | Ratio of Cost: DC - EC |
|---|---|
| 80 | 3:1 |
| 112 | 6:1 |
| 128 | 10:1 |
| 192 | 32:1 |
| 256 | 64:1 |

Freaking amazing stuff. Obviously, however, there is a slight calculation over head as computation over the EC can get expensive.

**Excersises From Chapter 1 of An Introduction to Mathematical Cryptography**

```
1.7
---

a)
    34787|353 = 353*q + r
```

```
    352*98 + 193
b)
    238792|7843 = 7843*q + r
    7843*30 + 3502
c)
    9829387493|873485 = 873485*q + r
    873485*11253 + 60788
d)
    1498387487|76348 = 76348*q + r
    76348*19625 + 57987

1.8
---

a)
    78745 (mod 127) = 5
b)
    2837647 (mod 4387) = 3645
c)
    8739287463 (mod 18754) = 17233
d)
    4536782793 (mod 9784537) = 6542162

1.9
---

a)
    291 = 252 * 1 + 39
    252 = 39 * 6 + 18
    39 = 18 * 2 + 3
    18 = 3 * 6 + 0
    gcd(291, 252) = 3
b)
    85652 = 16261 * 5 + 4347
    16261 = 4347 * 3 + 3220
    4347 = 3220 * 1 + 1127
    3220 = 1127 * 2 + 966
    1127 = 966 * 1 + 161
    966 = 161 * 6 + 0
    gcd(85652, 16261) = 161
c)
```

```
    139024789 = 93278890 * 1 + 45745899
    93278890 = 45745899 * 2 + 1787092
    45745899 = 1787092 * 25 + 1068599
    1787092 = 1068599 * 1 + 718493
    1068599 = 718493 * 1 + 350106
    718493 = 350106 * 2 + 18281
    350106 = 18281 * 19 + 2767
    18281 = 2767 * 6 + 1679
    2767 = 1679 * 1 + 1088
    1679 = 1088 * 1 + 591
    1088 = 591 * 1 + 497
    591 = 497 * 1 + 94
    497 = 94 * 5 + 27
    94 = 27 * 3 + 13
    27 = 13 * 2 + 1
    13 = 1 * 13 + 0
    gcd(139024789, 93278890) = 1
```

d)

```
    16534528044 = 8332745927 * 1 + 8201782117
    8332745927 = 8201782117 * 1 + 130963810
    8201782117 = 130963810 * 62 + 82025897
    130963810 = 82025897 * 1 + 48937913
    82025897 = 48937913 * 1 + 33087984
    48937913 = 33087984 * 1 + 15849929
    33087984 = 15849929 * 2 + 1388126
    15849929 = 1388126 * 11 + 580543
    1388126 = 580543 * 2 + 227040
    580543 = 227040 * 2 + 126463
    227040 = 126463 * 1 + 100577
    126463 = 100577 * 1 + 25886
    100577 = 25886 * 3 + 22919
    25886 = 22919 * 1 + 2967
    22919 = 2967 * 7 + 2150
    2967 = 2150 * 1 + 817
    2150 = 817 * 2 + 516
    817 = 516 * 1 + 301
    516 = 301 * 1 + 215
    301 = 215 * 1 + 86
    215 = 86 * 2 + 43
    86 = 43 * 2 + 0
    gcd(16534528044, 8332745927) = 43
```