Literature Survey of Post-Quantum Cryptography

Tanis Anderson[1*], Peterling Etienne[1†] and Michael Norberto[1†]

[1*]Department, Florida Atlantic University, 777 Glades Rd, Boca Raton, 33431, Florida, US.

*Corresponding author(s). E-mail(s): tanderson2018@fau.edu;
Contributing authors: petienne2020@fau.edu;
mnorberto2017@fau.edu;
†These authors contributed equally to this work.

Abstract

Current cryptography methods rely heavily on the how computationally infeasible it is to use brute force attacks on trapdoor functions. In particular, methods like RSA and ECC rely on the difficulty of three algorithms: the Integer Factorization, Discrete Log, and Elliptic-Curve Discrete Logarithm problems. However, recent advances in quantum computing mean that algorithms like Shor's Algorithm, that take advantage of quantum mechanics to quickly and efficiently make accurate guesses on prime factors and solve logarithmic problems, are able to break our current encryption methods in a matter of seconds. To rectify this, newer approaches to encryption are being found and used in what is called "post-quantum cryptography". These post-quantum cryptographic algorithms use many different methods of encryption, including complex polynomial math, stateless hash based approaches, and more. These new forms of encryption can provide the standards of protection against non-quantum attacks that are necessary, while also protecting against quantum attacks.

Keywords: post-quantum, cryptography, encryption, Shor's Algorithm, Grover's Algorithm

Summary

1 Introduction

We currently use RSA (Rivest Shamir Adleman) and ECC (Elliptic Curve Public-Key Cryptography) for a lot of our encryption methods. These methods are generally what is called public-key cryptography. In a public-key system, there are two keys used, one is called the public-key which is a specific number or formula used for encrypting messages, and the other is called the private key, which is generally an incredibly large prime number. Public-key cryptography works incredibly well because normal computers are not good at factoring prime numbers, which means that obtaining the private key is pointless in terms of computational time, power, and overall resources. For example, on a normal computer, a password comprised of 3763863863761 would take approximately 4 minutes to crack for a given computer, however, as we increase the length we can see that the time that it takes to crack increases exponentially. So if we just use that number two times in a row to get 376386386376137638638637 61, then the new time to crack the password becomes 79 million years. Obviously, there is more complexity to passwords than that, but it gives a good idea of how large prime numbers are harder for computers to crack. However, in 1994 a mathematician at MIT named Peter Shor came up with an algorithm that uses a lot of mathematical cleverness to create better guesses for the factors of a prime number. The algorithm works best in quantum systems, especially quantum systems with a higher number of what is referred to as qubits. In a normal computer system, the most basic unit of information is a bit, which represents a logical state where a value is either 1 or 0. Bit is actually an amalgamation of binary integer, and can be thought of as representing either an on or off state, and then using binary code and increasing layers of complexity, we get the computers that we use today. In a quantum system the most basic unit of information is not a bit, but a qubit (short for quantum bit). These qubits do not represent either on or off like a regular bit, but rather both numbers at once. Essentially, Shor's algorithm takes advantage of how qubits function to create a system that efficiently and effectively outputs the highest probability prime factors of a number. What would normally take current computers say 79 million years to crack, with Shor's algorithm it would take quantum computers a few minutes. This is obviously an incredibly big issue, because we use public-key cryptography all of the time, and so much of the internet, hardware, and infrastructure rely on these methods of encryption that would be rendered almost useless. Luckily, we still have time before reaching that point, because according to several estimates, for Shor's algorithm to be effective enough, it would require a system that has around 1300 to 1600 qubits, and currently the most advanced systems are only beginning to near 1000 qubits. The point of this project is to help update hardware for the transition period when Shor's algorithm does become a concern to encryption, and to open avenues for future quantum encryption methods in our software and hardware systems.

1.1 Motivations and Contributions

Cryptography is now a crucial component of every single interaction on the internet. It helps regulate and ensure the safety and privacy of everyday exchanges from financial information to medical information to sensitive government data. Secure cryptography is a necessary part of the internet and is used for billions of transactions and by billions of people. However, with the advance of quantum computers, these algorithms will become obsolete, and new algorithms for post-quantum cryptography will be necessary for the protection of user and governmental data. This paper looks at current solutions in the post-quantum cryptography space, and analyzes the necessity of them, as well as some of the challenges in implementing these solutions.

2 Background Information

2.1 Asymmetric Cryptography

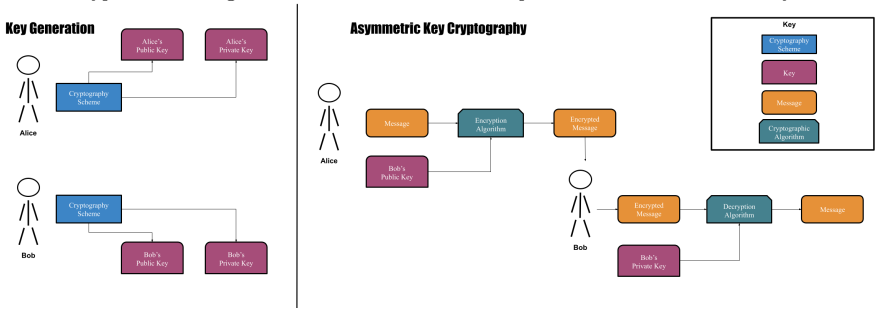
Asymmetric Cryptography is the backbone of our current encryption infrastructure and requires several pieces to function properly. When using asymmetric key cryptography, it can also be referred to as public-key cryptography, this is because the system uses a public key and a private key.

2.1.1 Public Key

The public key in a public-key system is generally a number generated by a cryptographic function for each user in a system. It is known by everyone, and is used to encrypt messages. For example, if Alice, wants to send a message to a different user, Bob, but doesn't want the information to be known to a malicious third party, then she can use Bob's public key, which is known to everyone, to encrypt the message. This does not compromise the security of the message, but means that it is now encrypted, and can in theory only be decrypted by Bob who has the private key that goes with his public key (Diffie).

2.1.2 Private Key

The private key in a public-key system is generally a number, or set of different numbers used in tandem, that correspond to a user's public key. Private keys are what make public-key systems work, they give the user the knowledge needed to take a message encrypted with their public key and decrypt it. Without the private-key it is computationally infeasible to decrypt the message. This is what's called a trapdoor function or a one-way function.



2.1.3 Trapdoor Functions

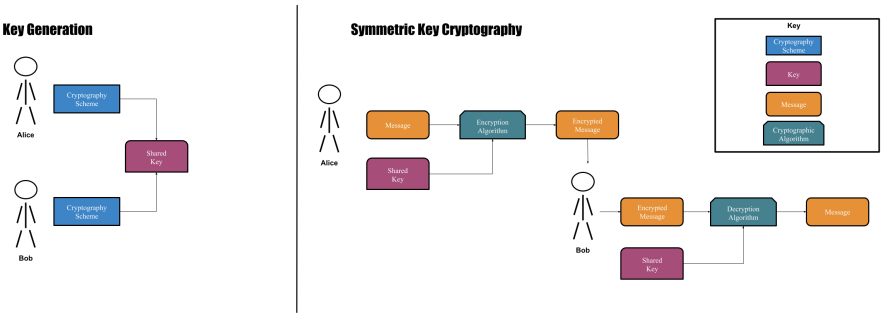
Trapdoor functions provide a major part of the foundation of modern cryptography. These functions are called trapdoor functions or one-way functions, because they work to take a message, perform an operation on it, and then obscure it to the point where it can't be undone. However, in cryptography, these functions have clever mathematics that allows a user with the right knowledge to undo the operation and get back to the original message. It is similar to taking 100 different pages of paper and shredding them. If a third party tries to put them back together, it will take them an unreasonable amount of time, however, since what was on each page is known to us, it is much faster, but still not instant, for us to piece the pages back together (Yao).

2.1.4 Putting it Together

In a public-key cryptosystem, there are several necessary pieces. There is the system itself, which is needed to generate the public and private keys, to encrypt and decrypt messages using trapdoor functions, and finally a signature scheme is necessary. Signature schemes are used to prove the authenticity of the message, and can be thought of like signatures on check. With all of these pieces, a system is capable of fully encrypting and decrypting messages and ensuring authenticity and message integrity.

2.2 Symmetric Cryptography

Symmetric Cryptography systems have been used in some form throughout history. Symmetric Systems unlike asymmetric systems require that there be a key in some form. That key is then shared between the two participating parties, and is then used for both encryption and decryption. One of the first examples was the Caesar Cipher, which was a basic shift cipher. This cipher used a simple alphabetic shift pattern, where the two participating parties would agree on how much to shift a letter. For example, shifting the alphabet over by one, so A becomes B and B becomes C and so on. This knowledge is required for both the encryption and decryption, and is needed by both parties for the Caesar Cipher, and symmetric systems generally, to be used correctly.



2.3 Quantum Cracking

As mentioned in the Introduction, with the increasing power of quantum computers, the feasibility of attacks against current computational methods is rapidly increasing. Conventional cryptography relies on the difficulty of three problems. The first problem is the Discrete Log Problem, the second is the Integer Factorization Problem, and the final is the Elliptic-Curve Discrete Logarithm Problem. All of these underlie current methods. For example, RSA encryption uses the difficulty of the Integer Factorization Problem to create the keys necessary for encryption and decryption, and ECC uses the difficulty of the Elliptic-Curve Discrete Logarithm Problem. However, Shor's Algorithm, came in part, as a solution to the Discrete Log Problem, and is able to rapidly and efficiently solve all 3 of the aforementioned problems using a quantum computer. This means that the main method of protection for many cryptosystems will be entirely obsolete in the face of operational quantum computers.

2.4 Post-Quantum Algorithms

Post-Quantum Algorithms are the necessary response to the increasing power of quantum computers. Due to the algorithms mentioned above like Shor's Algorithm and Grover's Algorithm, many of the most ubiquitous encryption methods will rapidly become obsolete. Post-Quantum Algorithms are cryptosystems that are specifically designed to be able to withstand both regular attacks and quantum ones. These algorithms generally use more complex mathematics, and focus on new solutions to avoid using prime number mathematics. For example, some systems use a polynomial lattice structure, which essentially limits the possible permutations of the polynomial after each mathematical operation. One example is the NTRUEncrypt algorithm, which uses trapdoor operations on randomly generated invertible polynomials to perform encryption and decryption (NTRUEncrypt Specifications). Other Post-Quantum Algorithms use modified versions of existing protocools. For example the SIKE algorithm uses Supersingular Isogeny Key mathematics. The algorithm uses a modification of the Diffie-Hellman algorithm, which has been a long-existing algorithm and would allow for easier implementation methods (SIDH Specifications). Other systems may use complex matrix mathematics, or other, newer methods.

3 Literature Survey

Within the past few decades encryption has become one of the most ubiquitous technologies in the world and is used and required for essentially all digital transactions. It is a crucial part of the security and privacy that many citizens have come to expect, and it is a crucial part of corporate and governmental data. However, while the necessity and ubiquity of encryption have increased in the past decades, the attacks against them have as well **(Insert Source Here)**. There is now an increasing threat by quantum computers to current encryption methods as established above. If this threat perseveres then the security and privacy of citizens, corporations, and governments will be at stake. Highly sensitive documents could be exposed, military, traffic, and energy systems could all be open to more frequent and dangerous attacks because of it **(Add Source Here)**. Organizations like NIST and governments are working to find alternatives to current methods of encryption that are resistant to quantum and classical attacks.

There are many issues that quantum-safe algorithms face. This can be computational feasibility as established in **(Add Source Here)**. Another point brought up by **(Add Source Here)**. However, a possible solution is presented in **(Add Source Here)**.

Many algorithms are using some variation of a few larger categories. Some are using modified and updated versions of classic encryption algorithms like **(Add Source Here)**. Or **(Add Source Here)** which updates **(Add Source Here)**.

Other algorithms are using a lattice structure which is not a well tested structure but appears to have good resiliency against classic and quantum attacks. **(Add Source Here)** does a good job combating the computational difficulty of classic lattice structures, but **(Add Source Here)** seems to be more flexible and capable of providing more use in complex situations. Although **(Add Source Here)** could be a good alternative to both as it provides a better balance.

4 Tables and Other Figures

5 Comparisons

The input format for the above table is as follows:

```
\begin{figure}[<placement-specifier>]
\centering
\includegraphics{<eps-file>}
\caption{<figure-caption>}\label{<figure-label>}
\end{figure}
```

Fig. 1 This is a widefig. This is an example of long caption this is an example of long caption this is an example of long caption this is an example of long caption

In case of double column layout, the above format puts figure captions/images to single column width. To get spanned images, we need to provide `\begin{figure*} ... \end{figure*}`.

For sample purpose, we have included the width of images in the optional argument of `\includegraphics` tag. Please ignore this.

6 Algorithms, Program codes and Listings

Packages `algorithm`, `algorithmicx` and `algpseudocode` are used for setting algorithms in LaTeX using the format:

```
\begin{algorithm}
\caption{<alg-caption>}\label{<alg-label>}
\begin{algorithmic}[1]
. . .
\end{algorithmic}
\end{algorithm}
```

You may refer above listed package documentations for more details before setting `algorithm` environment. For program codes, the "program" package is required and the command to be used is `\begin{program} ... \end{program}`. A fast exponentiation procedure:

```
begin
  for i := 1 to 10 step 1 do
      expt(2, i);
      newline() od                    Comments will be set flush to the right margin
where
proc expt(x, n) ≡
  z := 1;
  do if n = 0 then exit fi;
```

```
    do if odd(n) then exit fi;
       comment: This is a comment statement;
       n := n/2; x := x * x od;
    {n > 0};
    n := n - 1; z := z * x od;
  print(z).
end
```

Algorithm 1 Calculate $y = x^n$

Require: $n \geq 0 \vee x \neq 0$
Ensure: $y = x^n$

$y \Leftarrow 1$
if $n < 0$ **then**
  $X \Leftarrow 1/x$
  $N \Leftarrow -n$
else
  $X \Leftarrow x$
  $N \Leftarrow n$
end if
while $N \neq 0$ **do**
  if N is even **then**
    $X \Leftarrow X \times X$
    $N \Leftarrow N/2$
  else[N is odd]
    $y \Leftarrow y \times X$
    $N \Leftarrow N - 1$
  end if
end while

Similarly, for `listings`, use the `listings` package. `\begin{lstlisting} ... \end{lstlisting}` is used to set environments similar to `verbatim` environment. Refer to the `lstlisting` package documentation for more details.

```
for i:=maxint to 0 do
begin
{ do nothing }
end;
Write('Case insensitive ');
Write('Pascal keywords.');
```

7 Cross referencing

Environments such as figure, table, equation and align can have a label declared via the `\label{#label}` command. For figures and table environments use the `\label{}` command inside or just below the `\caption{}` command. You can then use the `\ref{#label}` command to cross-reference them. As an example, consider the label declared for Figure 1 which is `\label{fig1}`. To cross-reference it, use the command `Figure \ref{fig1}`, for which it comes up as "Figure 1".

To reference line numbers in an algorithm, consider the label declared for the line number 2 of Algorithm 1 is `\label{algln2}`. To cross-reference it, use the command `\ref{algln2}` for which it comes up as line 2 of Algorithm 1.

7.1 Details on reference citations

Standard LaTeX permits only numerical citations. To support both numerical and author-year citations this template uses `natbib` LaTeX package. For style guidance please refer to the template user manual.

Here is an example for `\cite{...}`: [1]. Another example for `\citep{...}`: [2]. For author-year citation mode, `\cite{...}` prints Jones et al. (1990) and `\citep{...}` prints (Jones et al., 1990).

All cited bib entries are printed at the end of this article: [3], [4], [5], [6], [7], [8], [9], [10], [11] and [12].

8 Examples for theorem like environments

For theorem like environments, we require `amsthm` package. There are three types of predefined theorem styles exists—`thmstyleone`, `thmstyletwo` and `thmstylethree`

`thmstyleone`	Numbered, theorem head in bold font and theorem text in italic style
`thmstyletwo`	Numbered, theorem head in roman font and theorem text in italic style
`thmstylethree`	Numbered, theorem head in bold font and theorem text in roman style

For mathematics journals, theorem styles can be included as shown in the following examples:

Theorem 1 (Theorem subhead) *Example theorem text. Example theorem text. Example theorem text. Example theorem text. Example theorem text. Example theorem text. Example theorem text. Example theorem text. Example theorem text. Example theorem text.*

Sample body text. Sample body text. Sample body text. Sample body text. Sample body text. Sample body text. Sample body text. Sample body text.

Proposition 2 *Example proposition text. Example proposition text. Example proposition text. Example proposition text. Example proposition text. Example proposition text. Example proposition text. Example proposition text. Example proposition text. Example proposition text.*

Sample body text. Sample body text. Sample body text. Sample body text. Sample body text. Sample body text. Sample body text. Sample body text.

Example 1 Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem.

Sample body text. Sample body text. Sample body text. Sample body text. Sample body text. Sample body text. Sample body text. Sample body text.

Remark 1 Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem.

Sample body text. Sample body text. Sample body text. Sample body text. Sample body text. Sample body text. Sample body text. Sample body text.

Definition 1 (Definition sub head) Example definition text. Example definition text. Example definition text. Example definition text. Example definition text. Example definition text. Example definition text. Example definition text.

Additionally a predefined "proof" environment is available: `\begin{proof} ... \end{proof}`. This prints a "Proof" head in italic font style and the "body text" in roman font style with an open square at the end of each proof environment.

Proof Example for proof text. Example for proof text. Example for proof text. Example for proof text. Example for proof text. Example for proof text. Example for proof text. Example for proof text. Example for proof text. □

Sample body text. Sample body text. Sample body text. Sample body text. Sample body text. Sample body text. Sample body text. Sample body text.

Proof of Theorem 1 Example for proof text. Example for proof text. Example for proof text. Example for proof text. Example for proof text. Example for proof text. Example for proof text. Example for proof text. Example for proof text. Example for proof text. □

For a quote environment, use `\begin{quote}...\end{quote}`

> Quoted text example. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

Sample body text. Sample body text. Sample body text. Sample body text. Sample body text (refer Figure 1). Sample body text. Sample body text. Sample body text (refer Table **??**).

9 Methods

Topical subheadings are allowed. Authors must ensure that their Methods section includes adequate experimental and characterization data necessary for others in the field to reproduce their work. Authors are encouraged to include RIIDs where appropriate.

Ethical approval declarations (only required where applicable) Any article reporting experiment/s carried out on (i) live vertebrate (or higher invertebrates), (ii) humans or (iii) human samples must include an unambiguous statement within the methods section that meets the following requirements:

1. Approval: a statement which confirms that all experimental protocols were approved by a named institutional and/or licensing committee. Please identify the approving body in the methods section
2. Accordance: a statement explicitly saying that the methods were carried out in accordance with the relevant guidelines and regulations

3. Informed consent (for experiments involving humans or human tissue samples): include a statement confirming that informed consent was obtained from all participants and/or their legal guardian/s

If your manuscript includes potentially identifying patient/participant information, or if it describes human transplantation research, or if it reports results of a clinical trial then additional information will be required. Please visit (https://www.nature.com/nature-research/editorial-policies) for Nature Portfolio journals, (https://www.springer.com/gp/authors-editors/journal-author/journal-author-helpdesk/publishing-ethics/14214) for Springer Nature journals, or (https://www.biomedcentral.com/getpublished/editorial-policies#ethics+and+consent) for BMC.

10 Discussion

Discussions should be brief and focused. In some disciplines use of Discussion or 'Conclusion' is interchangeable. It is not mandatory to use both. Some journals prefer a section 'Results and Discussion' followed by a section 'Conclusion'. Please refer to Journal-level guidance for any specific requirements.

11 Conclusion

Conclusions may be used to restate your hypothesis or research question, restate your major findings, explain the relevance and the added value of your work, highlight any limitations of your study, describe future directions for research and recommendations.

In some disciplines use of Discussion or 'Conclusion' is interchangeable. It is not mandatory to use both. Please refer to Journal-level guidance for any specific requirements.

Supplementary information. If your article has accompanying supplementary file/s please state so here.

Authors reporting data from electrophoretic gels and blots should supply the full unprocessed scans for key as part of their Supplementary information. This may be requested by the editorial team/s if it is missing.

Please refer to Journal-level guidance for any specific requirements.

Acknowledgments. Acknowledgments are not compulsory. Where included they should be brief. Grant or contribution numbers may be acknowledged.

Please refer to Journal-level guidance for any specific requirements.

Declarations

Some journals require declarations to be submitted in a standardised format. Please check the Instructions for Authors of the journal to which you are submitting to see if you need to complete this section. If yes, your manuscript must contain the following sections under the heading 'Declarations':

- Funding
- Conflict of interest/Competing interests (check journal-specific guidelines for which heading to use)
- Ethics approval
- Consent to participate
- Consent for publication
- Availability of data and materials
- Code availability
- Authors' contributions

If any of the sections are not relevant to your manuscript, please include the heading and write 'Not applicable' for that section.

Editorial Policies for:

Springer journals and proceedings: https://www.springer.com/gp/editorial-policies

Nature Portfolio journals: https://www.nature.com/nature-research/editorial-policies

Scientific Reports: https://www.nature.com/srep/journal-policies/editorial-policies

BMC journals: https://www.biomedcentral.com/getpublished/editorial-policies

Appendix A Section title of first appendix

An appendix contains supplementary information that is not an essential part of the text itself but which may be helpful in providing a more comprehensive understanding of the research problem or it is information that is too cumbersome to be included in the body of the paper.

References

[1] Campbell, S.L., Gear, C.W.: The index of general nonlinear DAES. Numer. Math. **72**(2), 173–196 (1995)

[2] Slifka, M.K., Whitton, J.L.: Clinical implications of dysregulated cytokine production. J. Mol. Med. **78**, 74–80 (2000). https://doi.org/10.1007/s001090000086

[3] Hamburger, C.: Quasimonotonicity, regularity and duality for nonlinear systems of partial differential equations. Ann. Mat. Pura. Appl. **169**(2), 321–354 (1995)

[4] Geddes, K.O., Czapor, S.R., Labahn, G.: Algorithms for Computer Algebra. Kluwer, Boston (1992)

[5] Broy, M.: Software engineering—from auxiliary to key technologies. In: Broy, M., Denert, E. (eds.) Software Pioneers, pp. 10–13. Springer, New York (1992)

[6] Seymour, R.S. (ed.): Conductive Polymers. Plenum, New York (1981)

[7] Smith, S.E.: Neuromuscular blocking drugs in man. In: Zaimis, E. (ed.) Neuromuscular Junction. Handbook of Experimental Pharmacology, vol. 42, pp. 593–660. Springer, Heidelberg (1976)

[8] Chung, S.T., Morris, R.L.: Isolation and characterization of plasmid deoxyribonucleic acid from Streptomyces fradiae. Paper presented at the 3rd international symposium on the genetics of industrial microorganisms, University of Wisconsin, Madison, 4–9 June 1978 (1978)

[9] Hao, Z., AghaKouchak, A., Nakhjiri, N., Farahmand, A.: Global integrated drought monitoring and prediction system (GIDMaPS) data sets. figshare https://doi.org/10.6084/m9.figshare.853801 (2014)

[10] Babichev, S.A., Ries, J., Lvovsky, A.I.: Quantum scissors: teleportation of single-mode optical states by means of a nonlocal single photon. Preprint at https://arxiv.org/abs/quant-ph/0208066v1 (2002)

[11] Beneke, M., Buchalla, G., Dunietz, I.: Mixing induced CP asymmetries in inclusive B decays. Phys. Lett. **B393**, 132–142 (1997) arXiv:0707.3168 [gr-gc]

[12] Stahl, B.: DeepSIP: Deep Learning of Supernova Ia Parameters, 0.42, Astrophysics Source Code Library (2020), ascl:2006.023