Quantum Cybersecurity Readiness

Mary Grace Rivero, Minh Tu, Saloni, Ruman Munir, Jericho Regala, Talha Mohammed Altamash Ghori

# Table of Contents

[Table of Contents 1](#_Toc133161452)

[I. Introduction 2](#_Toc133161453)

[II. Strategic Themes 3](#_Toc133161454)

[III. Roadmap 4](#_Toc133161455)

[Year-1: Removal of Weak Ciphers and Enforcing Strong Ciphers 5](#_Toc133161456)

[Bow Valley College Weak Ciphers Removal Process 5](#_Toc133161457)

[1. Assessing & Prioritizing 6](#_Toc133161458)

[A.TLS Protocol 7](#_Toc133161459)

[B. Key Exchange 9](#_Toc133161460)

[C. Bulk Encryption Ciphers 12](#_Toc133161461)

[D. Hashing Algorithms 17](#_Toc133161462)

[2. Testing 20](#_Toc133161463)

[3. Deploying 21](#_Toc133161464)

[4. Monitoring 21](#_Toc133161465)

[Summary of Year-1 Recommendations 21](#_Toc133161466)

[Year-2: Anticipatory Planning for Post-Quantum Break 23](#_Toc133161467)

[NIST Candidates for Standardization 25](#_Toc133161468)

[1. CRYSTALS-KYBER 25](#_Toc133161469)

[2. CRYSTALS-DILITHIUM 26](#_Toc133161470)

[3. FALCON 27](#_Toc133161471)

[4. SPHICS+ 28](#_Toc133161472)

[Other Post-Quantum Ciphers that can be considered 29](#_Toc133161473)

[Summary of Year-2 Recommendations 30](#_Toc133161474)

[IV. Conclusion 31](#_Toc133161475)

[V. References 32](#_Toc133161476)

# Introduction

In recent years, the field of cryptography has experienced a paradigm shift with the advent of quantum computers. The advances in quantum computing pose a significant threat to the security of the classical cryptographic systems that are widely used today. It is imperative to upgrade the existing cryptographic systems and develop post-quantum cryptography to protect against quantum attacks. In this paper, the team will develop a two-year roadmap for Bow Valley College, to somehow prepare for the post-quantum break that is to come.

# Strategic Themes

The strategic themes that will be followed for the 2-year road map are as follows:

|  |
| --- |
| **Strategic Theme 1 -** Compliance and Integration: Removal of Weak Ciphers and Enforcing Strong Ciphers |
| **Strategic Theme 2 –** Futureproofing: Anticipatory Planning for Post-Quantum Break |

The IT systems at Bow Valley College incorporates numerous cipher suites that are susceptible to security attacks, particularly from quantum computers. The initial strategic objective of this research paper is Compliance and Integration which involves eliminating the usage of weak ciphers as a precautionary measure in anticipation of the post-quantum era. With classical computers still employing obsolete and insecure ciphers, it is imperative to safeguard them against current attacks before even considering post-quantum computers. Therefore, it is vital to attain a level of security that can withstand attacks from classical computers before attempting to address a future threat that might still be decades away. The elimination of weak ciphers ensures that only strong ciphers remain, thereby fortifying the system against potential cyber threats. The strong ciphers, which conform to the current National Institute of Standards and Technology (NIST) standards, are state-of-the-art encryption techniques, and their use ensures the organization's compliance with regulatory standards as well as ensuring the security of the organization with the deployment of the current approved standards. Keeping up with the current secure standards gives Bow Valley College one step further toward the transition of being post-quantum resilient. Testing the compatibility of new protocols with existing systems and identifying any necessary changes or upgrades to ensure smooth integration must be done as part of the first strategic theme.

As for the second strategic objective, transitioning to post-quantum cryptography would be a difficult task as Chen et al. (2016) have stated but preparing for it will never be too early. Currently, as of writing, there are four (4) candidates that the National Institute of Standards and Technology has chosen and analyzed to be quantum-resilient, and which are expected to become part of the post-quantum cryptographic standard. Introducing these four and how they can be implemented in a hybrid approach would be the focus of the second strategic theme which is anticipatory planning for the post-quantum break.

# Roadmap

A screenshot of a computer

Description automatically generated with medium confidence

Figure :Roadmap Overview

Plans for the first year would mainly focus on strategic theme 1, which is compliance and integration. It involves removing weak ciphers and enforcing the strong ones, ensuring compliance and integration of the systems. Year-1 plans will start on May 1, 2023, and it is estimated to end on April 28, 2024.

As for year 2, it will mainly focus on strategic theme 2 which is futureproofing, entailing preparations and anticipatory planning for the post-quantum break. Year-2 plans will start on May 1, 2024, and it is estimated to end on April 25, 2025. Where all planning and scheduling are concerned, these are rough estimates that can become shorter or longer, depending on the results of each phase.

## Year-1: Removal of Weak Ciphers and Enforcing Strong Ciphers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Year | Start | End | Duration (Days) | Label |
| First Year | **Removal of weak Ciphers/Apply control if cannot remove weak Cipher** | | | |
| 5/1/2023 | 5/30/2023 | 30 | **Finding & Priority Phase**  **May 1 - May 30** |
| 5/31/2023 | 7/29/2023 | 60 | **Testing Phase**  **May 31 - Jul 29** |
| 7/30/2023 | 9/27/2023 | 60 | **Deployment Phase**  **Jul 30 - Sep 27** |
| 9/28/2023 | 10/27/2023 | 30 | **Monitoring Phase**  **Sep 28 - Oct 27** |
| **Improving Current Cipher to Latest Algorithm** | | | |
| 11/1/2023 | 11/30/2023 | 30 | **Improvement Phase**  **Nov 1 - Nov 30** |
| 12/1/2023 | 1/29/2024 | 60 | **Testing Phase**  **Dec 1 - Jan 29** |
| 1/30/2024 | 3/29/2024 | 60 | **Deployment Phase**  **Jan 30 - Mar 29** |
| 3/30/2024 | 4/28/2024 | 30 | **Monitoring Phase**  **Mar 30 - Apr 28** |

Table 1: Breakdown of Schedule for Year-1

Considering the time and effort required to remove the weak ciphers and ensure compatibility with the remaining ones, a comprehensive and systematic approach is necessary to ensure a smooth transition. Thus, a four-phased plan will be employed to guarantee that the system is compatible with the new set of cipher standards. The phases are as follows: Assessing and Prioritizing, Testing, Deploying, and Monitoring.

### Bow Valley College Weak Ciphers Removal Process

##### 1. Assessing and Prioritizing

The first phase of the plan involves a thorough evaluation of the system to identify any weak ciphers that must be replaced or removed entirely. This includes classifying VLANs, and Domains based on their importance and determining the number of servers impacted by weak ciphers. This phase is expected to take about one month.

##### 2. Testing

Before deploying the new cipher suites, it is critical to test them in virtual settings to guarantee that their removal or replacement does not cause system issues. This testing phase is planned to last around a month. There will be a couple tests that will be done to ensure that the changes in place are working and are stable. The main two tests that will be done will be testing on the research and development environment and the second test being in a virtual machine environment.

##### 3. Deploying

The new cipher suites can be deployed once the testing phase is complete, and it has been confirmed that all systems are running properly with just the approved ciphers. This will be a gradual procedure to ensure that server performance is not affected. In addition, we could create an image with correct config to apply for other systems. The deployment phase should take around two months.

##### 4. Monitoring

After the new cipher suites have been implemented, it is critical to regularly monitor the system for any difficulties that may develop. Continuing strict monitoring shall be done, checking that everything is working, without encountering any problems for at least a month. When the systems are stable and there are not any encountered problems within a month, that can signify a lessened monitoring schedule, but logging will be kept up to date to discover any unforeseen difficulties.

The BVC weak ciphers removal process is a recursive process that will be implemented gradually in a phased approach, which will be one VLAN at a time.

For the first half year, the focus is all about improving the currently employed cipher suites to comply with the NIST standard by ensuring that the recommended protocols are used or prioritized. As part of the improvement, removing the weak ciphers and finding some ways to improve the cipher suites will be planned and implemented.

### 1. Assessing & Prioritizing

In preparation for the post-quantum project, Bow Valley College’s IT Security Team prepared an inventory of the TLS protocols, key exchanges, ciphers, hashing algorithms, and cipher suites currently deployed in the IT infrastructure at Bow Valley College which can be found in the attached excel file. The Excel sheet was assessed by looking at the security of the ciphers used and below are the overall findings.

Figure : BVC Cipher Suites Security Level [Redacted]

Based on the inventory of ciphers currently deployed in Bow Valley College, [Redacted]. Further discussion as to how the ciphers were assessed can be below:

#### A.TLS Protocol

TLS stands for Transport layer Security which is a cryptographic protocol widely used in sending emails, instant messages, and securing HTTPS. TLS is an upgraded version of SSL. The TLS and SSL protocols are designed to facilitate protection to data and communication over the internet securely. It aims primarily to provide security, including CIA using cryptography. TLS has evolved over time to keep up with the complex cryptographic protocols. The different TLS versions were introduced over time, and the latest one was TLS 1.3 in August 2018. TLS1.2 is still prevalent today, but NIST stated that by January 1, 2024, all systems must be able to support TLS1.3 (McKay & Cooper, 2019).

###### Importance of removal of older SSL, TLS 1.0 and 1.1

The Internet Engineering Task force (IETF) has stated very specifically that there will be no use of TLS1.0 and TLS1.1 as they want to ensure the use of secure protocol for transactions.

TLS 1.0 was first deployed in January 1999 and has replaced the SSL protocol. It has been deprecated and in which having this protocol would make any retailer non-compliant with the Payment Card Industry Data Security Standard (PCI DSS) standard.

Both protocols have various vulnerabilities such as:

* Both TLS 1.0 and TLS 1.1 depend on running SHA-1 and MD-5 hash to exchanged messages which is a weak hash. This will make room for an attacker to perform a downgrade attack on the handshake.
* The process of the handshake totally depends on digital signatures and using weak hashes like SHA-1 hash or MD-5 (Message Digest 5) allows the attacker to impersonate a server.
* No older versions of protocols like SSL and TLS allow the user to select a stronger hash for signatures. The only way is to use the latest protocol version.
* Using the outdated version can lead to phasing out support and can increase the attack surface and the scope of maintenance. (Diot et al., 2000)

###### Improvements brought by TLS 1.2 and 1.3

TLS 1.2 was first released in 2008. It offers more security and designs for both high performance and improved reliability. To accomplish this, we need both cryptography combination of symmetric and asymmetric. Some of the things that were introduced in the TLS 1.2 are Advanced Encryption Standard (AES) for encryption, replacing MD5/SHA-1 combination in the digitally signed element with a single hash, and an improved padding scheme that removes the previous vulnerability to BEAST attack. In addition, TLS 1.2 is becoming widely popular for providing secure online transactions. TLS 1.1 has been deprecated by many organizations as it fails to provide enough security. TLS 1.2 has now become mandatory for security standards and regulations, such as PCI. In comparison to TLS 1.3, TLS 1.3 is the latest version that does not support old cryptographic security standards. TLS 1.3 protocol has many advantages over its older version such as increased performance, security, and zero round trip time, which makes it most promising to be implemented in today’s time. (Armbrust et al., 2010)

##### TLS Findings

Looking at the inventory of Bow Valley College, it still supports [Redacted] for all VLANS. It also has [Redacted] which are considered more vulnerable than the [Redacted] protocols. All the older versions of SSL and TLS protocols are discovered with many security threats and loopholes. For example, Lucky Thirteen attack which exploits CBC, BEAST attack which exploits the padding scheme, and POODLE. Having vulnerabilities like these can cause leak of data and information and to avoid that we should not be using SSL and TLS 1.0 and 1.1. Table 2 below explains all the vulnerabilities found in TLS 1.0 and 1.1: (Cho et al., 2019)

|  |  |  |
| --- | --- | --- |
| Encryption Algorithm | Vulnerability | CVE |
| TLS1.0 | Padding error | CVE-2019-3730: Vulnerable to an information exposure (*NVD - CVE-*2019-3730, n.d.) |
|  | Initialization  vector selection | CVE-2021-26407: May lead to a collision of IVs with the same key potentially resulting in information disclosure (*NVD - CVE-*2021-26407, n.d.) |
| TLS 1.1 | BEAST Attack | CVE-2011-3389: This allows man-in-the-middle to obtain plain text (*NVD - CVE-*2011-3389, n.d.) |
|  | RC4 Attack | CVE-2013-2566: RC4 makes it easy for attackers to recover the plain text (*NVD - CVE-*2013-2566, n.d.) |
|  | Attacks renegotiation | CVE-2009-3555: No proper establishment of handshake which allows MITM to insert data in between session(*NVD - CVE-*2009-3555, n.d.) |

Table 2: TLS vulnerabilities

##### TLS Recommendations

With the above discussion, we can recommend some changes and implement using the modern protocol. Use of the latest versions like TLS 1.2 and 1.3, they are more reliable, faster and are more secure than the older version. The one update done in TLS 1.3 is the handshake process in which it only requires one round trip instead of two which has shortened the process in compared to other TLS protocols. This helps to make the HTTPS connections faster and stronger and makes improvements to the overall user experience. (Diot et al., 2000)

#### B. Key Exchange

The secure transmission of data relies heavily on cryptography, which secures sensitive information from unwanted access. Secure communication channels between two parties are facilitated through key exchange algorithms that form a fundamental component of cryptography. But the increase and progress in the use of quantum computing technology have made a massive impact and as a result, threats are constantly increasing.

The key exchange algorithms discussed in the provided Excel file, such as ECDHE, ECDH, RSA, DH, and PSK, may become susceptible to attacks leading to data breaches and the loss of privacy.

As a result, we are proposing 2 years road map which examines the need to improve the resilience of the key exchange ciphers against quantum threats through their removal or replacement.

##### Key Exchange Findings

###### Ephemeral Elliptic Curve Diffie-Hellman (ECDHE)

The Ephemeral Elliptic Curve Diffie-Hellman algorithm is a commonly utilized technique for secure communication due to its ability to offer flawless forward secrecy and withstand passive eavesdropping. This is also used in SSL/TLS protocols. However, ECDHE has a weakness that leaves it vulnerable to quantum attacks. These attacks use Shor's algorithm, which can efficiently factorize the large prime numbers used in ECDHE. Hence, attack vectors can rapidly recover the private key and obtain access to sensitive data.

In the process of the exchange of keys, ECDHE creates a new set of keys for every connection, with the private key being kept confidential, and only the public key that is shared with the other participant. An attack vector can't decrypt past or future messages, even if they intercept the communication and acquire the private key. This guarantees absolute confidentiality through perfect forward secrecy.

ECDHE is a strong key exchange algorithm that is not affected by quantum computers as of now. It can be considered resilient against brute-force attacks and attacks from adversaries with large computing power such as quantum computers. Therefore, there is no need to replace or remove ECDHE for quantum resiliency right now.

###### Elliptic Curve Diffie-Hellman (ECDH)

This is another key exchange algorithm that utilizes elliptic curve cryptography for secure communication channels. In contrast to ECDHE, ECDH is not as strong as it doesn't provide perfect forward secrecy. For example, consider an event where a threat actor gains access to the private key, they can decrypt past and future communications. Moreover, ECDH is not suitable for use in quantum-resistant communication systems because of its vulnerability to quantum attacks. Hence by these considerations ECDH should be replaced or removed for the sake of quantum resiliency.

###### Rivest-Shamir-Adleman (RSA)

Rivest-Shamir-Adlemancan be categorized as a widely used algorithm for public-key cryptography, used for encryption and digital signatures. It is based on the difficulty of factorizing large prime numbers, but it has been found vulnerable to Shor’s algorithm. With the advancement of quantum computing, RSA's security level is diminishing, and it is no longer a suitable option for quantum-resistant communication systems. Although RSA is frequently used in SSL/TLS protocols for key exchange, it is not a strong competitor against quantum attacks, and therefore, it is supposed to be replaced or removed to improve quantum resiliency.

###### Diffie-Hellman (DH)

The Diffie-Hellman (DH) algorithm is used to create a mutual confidential key between two parties. The DH algorithm depends on the simplicity of calculating the exponentiations of a numerical value and the complexity of knowing the logarithm of that value. In this situation, both sides create a set of keys using the DH algorithm, which includes a public key and a private key. The public keys are swapped between the parties, and a common secret key is derived using the respective private keys and the received public keys. It is considered a robust key exchange algorithm against classical computers but not secure against quantum computers as Shor's algorithm can efficiently solve the discrete logarithm problem. As a result, the removal or replacement of the DH algorithm is justified for quantum resiliency.

###### Pre-Shared Key (PSK)

A Pre-Shared Key (PSK) is not a key exchange algorithm but is used in wireless communication and VPNs, where both parties have a shared secret key. Unfortunately, PSK is susceptible to attacks by attack vectors that intercept the communication and obtain the key. After that, once the key is compromised, the attack vector will be able to decrypt all messages sent by using the same key, thereby making PSK vulnerable to both classical computers and more so from quantum computers. Hence, anything with PSK should be replaced or removed entirely to avoid it from being exploited.

##### Key Exchange Recommendations

To summarize the whole process of key exchange, it is an essential part of cryptography that enables secure communication between two parties. Nonetheless, the growing use of quantum computers represents a substantial danger to the protection of key exchange algorithms that depend on intricate mathematical problem-solving.

The algorithm ECDHE exhibits robustness against attacks from quantum computers, making it a reliable key exchange method while algorithms such as ECDH, RSA, DH, and PSK are considered vulnerable to attacks. Hence, it is crucial to replace or eliminate these algorithms with those that can resist quantum-related threats to secure communication channels in the future. Cipher suites that make use of ECDHE should be prioritized and the rest should either be removed if possible or replaced with ECDHE.

Other key exchanges that can be made use of are Quantum Key Distribution and Super singular Isogeny Diffie-Hellman (SIDH). If possible, it is recommended to replace ECDHE with QKD for better security against all types of attacks, including quantum attacks. For ECDH, SIDH should be used instead of ECDH, as it is resistant to known classical and quantum attacks. Post-quantum cryptography algorithms like Hash-based cryptography, Lattice-based cryptography, or Code-based cryptography should be adopted instead of RSA and DH to improve quantum resilience. Finally, replacing the PSK with QKD can also enhance quantum resilience for secure communication channels. All the quantum resilient key exchanges mentioned above will be discussed further in the year-2 anticipatory planning for post-quantum resiliency.

#### C. Bulk Encryption Ciphers

In a cipher suite, bulk encryption cipher is composed of an encryption algorithm, encryption key size, and encryption mode. Each portion contributes to making the encryption stronger, thereby increasing the security of the communication between the client and the server that uses a particular cipher suite.

###### Encryption Algorithm and Key Size

The Encryption algorithm and the key size determine the process of transforming the messages exchanged between clients and servers to an unreadable form and are usually symmetric algorithms as it performs the process on a large amount of data. Symmetric encryption is used as it is faster and more efficient for performing bulk encryption as compared to asymmetric encryption which is slower and requires more resources (Ferguson, N., Schneier, B., & Kohno, T., 2010).

There are a couple of encryption algorithms that are present in the cipher suites being used in the VLANs of Bow Valley College, with varying key sizes. A summary can be found in the table below:

|  |  |
| --- | --- |
| Encryption Algorithm | Key Size/Strength |
| Data Encryption Standard (DES) | 56 |
| Triple Data Encryption Standard (3DES) | 168 |
| Rivest Cipher 2 (RC2) | 40 | 56 | 128 |
| Rivest Cipher 4 (RC4) | 40 | 56 | 64 | 128 |
| SEED | 128 |
| International Data Encryption Algorithm (IDEA) | 128 |
| Advanced Encryption Standard (AES) | 128 | 256 |
| CHACHA/POLY1305 | 256 |
| Camellia | 128 |

Table 3: Encryption Algorithms in BVC

###### Encryption mode

Encryption mode is the mode of operation being used in conjunction with the encryption algorithm. The two broad types of encryption techniques are Block Cipher and Stream Cipher. When a message is encrypted in a fixed-size data block, the technique is categorized as a block cipher while Stream cipher employs dynamic modifications on the input data to encrypt a constant sequence of binary digits (Difference Between Block Cipher and Stream Cipher - Javapoint, n.d.). This encryption technique operates on a per-bit basis, utilizing a stream of key bits to generate cipher text for any length of plain text messages. There are generally five modes of operation which are Electronic Codebook (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR) modes which provide the confidentiality of the communication (Dworkin, M.J., 2007). They were originally recommended by NIST but in May 2021, NIST initiated a review process of the recommendation as all five modes were found to have vulnerabilities (Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, U.S. Department of Commerce, n.d.).

Later in 2007, NIST announced the release of a new mode of operation which is Galois/Counter Mode (GCM) which was designed and proposed by McGrew and Viega (Dworkin, 2007).In the journal article by Dworkin (2007), it stated that GCM is similar to CTR except it provides a message authentication code (MAC) that provides assurance of confidentiality and authenticity of confidential data as it can detect both accidental modifications of data and unauthorized modifications. GCM is a stronger and much more secure encryption type that is no longer a block or stream kind, but a new class called an Authenticated Encryption with Associated Data (AEAD), which is what is being used for all TLS 1.3 cipher suites.

Encryption modes used in Bow Valley College are the following:

|  |  |
| --- | --- |
| Encryption Mode | Encryption Type |
| Cipher Block Chaining (CBC) | Block |
| CTR + CBC-MAC (CCM) | Block |
| Galois/Counter Mode (GCM) | Block |

Table 4: Encryption modes in BVC

##### Bulk Encryptions Ciphers Findings

Looking at the cipher suites, most of the suites that employ DES, 3DES, RC2, and RC4 are considered weak or insecure ciphers, regardless of their key size. This is due to the fact the DES and RC algorithms are deprecated because their vulnerabilities, when exploited, can cause a breach of confidentiality and integrity. Below are some of the vulnerabilities present in each encryption algorithm:

|  |  |  |
| --- | --- | --- |
| Encryption Algorithm | Vulnerability | CVE |
| DES | * Differential Cryptanalysis * Brute-force attack * Linear Cryptanalysis * Sweet32 Attack | * CVE-2016-2183: DES and 3DES birthday bound vulnerability that allows for remote attackers to obtain cleartext data (*NVD - CVE-2016-2183*, n.d.) |
| 3DES | * Sweet32 Attack | * CVE-2016-2183: DES and 3DES birthday bound vulnerability that allows for remote attackers to obtain cleartext data (*NVD - CVE-2016-2183*, n.d.) |
| RC2 | * Related-key Cryptanalysis |  |
| RC4 | * Key Scheduling Algorithm Weaknesses * Bar Mitzvah Attack | * CVE-2013-2566: Single-byte biases that makes it easier for remote attackers to recover plaintext (*NVD - CVE-2013-2566*, n.d.) * CVE-2015-2808: Exploits in RC4 that makes it easier for remote attackers to recover plaintext through its Invariance Weakness (*NVD - CVE-2015-2808*, n.d.) |

Table 5: Encryption Algorithms in BVC

Moreover, among the list of encryption modes used in BVC, what is noticeable is that most of the cipher suites use CBC or GCM. CBC has several vulnerabilities as well. Some of the most notable vulnerabilities of CBC are the BEAST attack which exploits TLS version 1.0 and older SSL protocols and the POODLE attack which exploits the SSL3.0 protocols (Kiprin, 2022). A summary of vulnerabilities and some CVEs related to CBC can be found in Table 4: CBC Vulnerabilities.

|  |  |
| --- | --- |
| CBC Vulnerability | CVE |
| Browser Exploit Against SSL/TLLS (BEAST) Attack | * CVE-2011-3389: Man-in-the-middle attack obtaining plaintext HTTP headers via BCBA in conjunctions with JavaScript code (*NVD - CVE-*2011-3389, n.d.) |
| Padding Oracle on Downgraded Legacy Encryption (POODLE) attack | * CVE-2014-3566: Exploits padding vulnerability that makes it easier for MITM attack to obtain cleartext data (*NVD - CVE-*2014-3566, n.d.) |
| Lucky Thirteen Attack | * CVE-2013-0169 – Timing side-channel attack against a MAC that allowed attackers to send specially crafted requests to the server that leads to unauthorized disclosure of sensitive information (*NVD - CVE-*2013-0169, n.d.) |

Table 6: CBC Vulnerabilities

Although GCM is quite secure, it should still be used with caution as Rogaway stated. GCM must operate under the assumption that the Initialization Vector (IV) should be unique— a nonce that should never be reused or repeated as a breach of this condition may result to forgery attacks (Dworkin, 2007).

##### Bulk Encryption Ciphers Recommendations

With the information above, the recommendations to meet the team’s strategic goals of removing weak ciphers and introducing a more secure ciphers for bulk encryption are as follows:

Disable or remove all cipher suites that use the weak ciphers (DES, 3DES, RC2, RC4) and ensure that cipher suites with AES\_256\_GCM, CHACHA20/Poly1305 are employed as they are some of the current strongest ciphers that are present in the BVC’s environment.

* Microsoft provided a link as to how RC4 can be disabled (*Microsoft Security Advisory: Update for Disabling RC4 - Microsoft Support*, n.d.).
* Disable or remove all cipher suites that use CBC as the mode of operation. Microsoft stated that if CBC mode is used, it must be incorporated with the use of secret-keyed data integrity check or change the cipher mode to an Authenticated Encryption (AE) mode such as GCM or CCM (Blowdart, 2022).
* If disabling or removing the cipher suite is not possible, a workaround can be providing an order for the cipher suite, ensuring that the cipher suites implementing strong ciphers that provide authenticated encryption such as GCM are prioritized. Microsoft has also provided a link on how to configure the Cipher Suite Order by using Group Policy (Andrei-Popov, 2023).
* Deland-Han (2023), and other contributors also provided information on how to deploy custom cipher suite ordering for a Windows Server 2016.
* If possible, use cipher suites using Authenticated Encryption with Associated Data (AEAD) which prevents all the CBC-based attacks mentioned in Table 6 as it is used for protecting confidentiality, integrity, and message authentication (McKay & Cooper, 2019). AEAD was introduced in TLS 1.2 and 1.3 protocols. Some examples are:
* CAMELLIA\_128\_GCM
* ARIA\_128\_GCM
* CHACHA20\_POLY1305
* AES\_128\_CCM

#### D. Hashing Algorithms

###### Hashing

Hashing is an algorithmic mathematical technique that scrambles a block of plain text message into a fixed unreadable irreversible block of text which is often used as an integrity check of the text message (AboutSSL, n.d.). In theory, hashing is meant to be irreversible using the hash function, but the evolution of traditional computers made past hashing algorithms vulnerable and irrelevant for securing plain text. Security professionals and day-to-day users at Bow Valley College use hashing algorithms provided by VLANS and third-party services. Hashing is used in Message Digest algorithms, SHA-1(Secure Hash Algorithm), SHA-2, and SHA-3 (Baeldung & Baeldung, 2023).

###### Message Authentication Code

Message Authentication Code or MAC is another method to encrypt hashing algorithms. This encryption process uses a symmetric key algorithm which adds a tag to the keys. The third party will not know what the original message is without the underlying shared symmetric shared secret key of a message authentication code. The tags in the keys provide security in the integrity and authentication of the key. Any changes made to the keys tag will result in a different type of MAC. An example of a MAC is Poly1305 which is seen in [Redacted], [Redacted], and [Redacted]. MAC can combine with hash algorithms such as SHA-1 and SHA-2 to create HMAC (Baeldung & Baeldung, 2023).

##### Hashing Algorithms Findings

The table below will show the hashing algorithms found in [Redacted], [Redacted], and [Redacted].

|  |  |
| --- | --- |
| Hash Algorithm | Key Length (bits) |
| MD5 (Message Digest 5) | 128 |
| SHA-1 | 160 |
| SHA 2 256 | 256 |
| SHA 2 384 | 384 |
| Poly1305 | 384 |

Table 7: Hash Algorithm's in BVC (Bow Valley College)

Below are some CVE (Common Vulnerabilities and Exposures) vulnerabilities found in SHA-1 and MD5:

|  |  |
| --- | --- |
| **Hash Algorithm** | CVE- Vulnerabilities |
| MD5 (Message Digest 5) | * CVE-2020-5229: Password Hash collision allowed threat actor to have the same admin login. (*NVD - CVE-*2020-5229, n.d.) * CVE-2004-2761: Spoofing attacks with X.509 digital certificate standard (*NVD - CVE-*2004-2761, n.d.) |
| SHA-1 (Secure Hash Algorithm) | * CVE-2005-4900: Failure of collision resilience making it easier to attack TLS (*NVD - CVE-*2005-4900, n.d.). |

Table 8: Hash Vulnerabilities

###### Secure Hash Algorithm

The secure hashing algorithm created by NIST is another variant of Message Digest 5 which is used to create digital certificates and hash data (Puneet, 2022). The secure Hash Algorithm version uses a method of a one-way hash function that results in a hashed message digest that is more secure than message digest version five. In recent years secure hash algorithm version is not as often used as a hashing algorithm as traditional computing attacks have become advanced making SHA-1 highly susceptible to brute force and collision attacks. The reason SHA-1 is vulnerable is due to its 160-bit fixed length that allows collision to occur. Collision in SHA-1 is the replication of similar hashed messages of the originally hashed message which violates integrity checks. Cipher suites that contain the first version of SHA should be removed or updated to the latest SHA-2 (“NIST Retires SHA-1 Cryptographic Algorithm | NIST,” 2022). Threat actors can create the same message digest as the original SHA-1 message digest and they can also duplicate the same certificate that the original SHA-1 has created with its message digest. Since SHA-1 is considered a weak hashing algorithm it is not cyber-resilient to quantum attacks or traditional computer attacks.

###### MD5 (Message Digest 5)

MD5 or Message Digest 5 is developed by Ronald Rivest contributor to the RSA (Rivest Shamir Adleman) algorithm. This hashing algorithm is the fifth variant of the cryptographic hashing algorithms of Message Digest. It is considered unsafe as it is highly susceptible to birthday attacks and collision attacks (Okta, n.d.). Any MD5 that is in the cipher suite should be removed immediately as it is not cyber resilient for both traditional computing attacks and quantum computing attacks.

###### Secure Hash Algorithm 2 (256 and 384)

Secure Hashing Algorithm 2 is the standard for certificates and hashing in today’s modern hashing algorithm (Lake & Lake, 2022). It is used widely throughout the internet and is supported and developed by NIST. SHA-2 has the same algorithmic method of hashing messages/passwords as SHA-1 with a difference of the bit size length. SHA-2 should not be removed from the current cipher suite for Bow Valley College VLANs is that it supports TLS, IPsec, S/MIME, SSH (Secure Shell), and PGP security protocol which can protect against traditional computer attacks. The attack in question are collision attacks, SHA-1 and MD5 are susceptible to collision attacks while SHA-2 solves that problem by increasing the length of the bits to 256 bits or 384 bits. SHA-2 is not one hundred percent protection against collision attacks as it is still susceptible to brute force attacks, another alternative to SHA-2 is SHA-3 but it is not used as widely as SHA-2.

###### POLY1305

POLY1305 is a message-authenticated code technique used once to authenticate messages using a symmetric key algorithm. It is used in TLS 1.3 cipher suite as the security of this MAC hash algorithm uses a 16-byte by 16-byte key and message (Langley & Nir, 2015). Additionally, the bytes are quite unique in 128-bit fixed that can flag a message for modification which can show that the bits have changed. POLY1305 is efficient and faster than most modern hashing algorithms which is why the cipher suite [Redacted] that resides in the Bow Valley College VLANs is quite secure.

##### Hashing Algorithms Recommendations

With the above-mentioned hashing algorithms and message authentication code, security teams can remove certain cipher suites that currently use SHA-1 and MD5. The exemption condition of the removal of cipher suites that use MD5, or SHA-1 is if certain VLANs do not function properly due to the removal of a cipher suite that contained SHA-1 or MD5 further inspection will be required to securely remove the cipher from the cipher suite. It is highly recommended that security professionals at Bow Valley College should use SHA-2 or SHA-3, or POLY1305 for hashing algorithms as they are less susceptible to hashing attacks.

Below are some examples from [Redacted], [Redacted], and [Redacted] which security teams can remove:

* TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
* TLS\_RSA\_WITH\_RC4\_128\_SHA
* ECDHE\_RSA\_AES256\_SHA
* TLS\_RSA\_WITH\_RC4\_128\_MD5
* SSL\_CK\_DES\_192\_EDE3\_CBC\_WITH\_MD5
* SRP\_RSA\_AES\_128\_CBC\_SHA
* SEED\_SHA
* ECDHE\_PSK\_AES256\_CBC\_SHA
* DHE\_RSA\_SEED\_SHA
* RC4\_MD5
* DH\_RSA\_AES128\_SHA

### 2. Testing

The testing phase encompasses the critical decision-making process of determining which VLANs to prioritize when applying the proposed changes. This prioritization procedure aims to enhance the effectiveness and efficiency of the testing process. Once the prioritized VLANs have been identified, the testing phase will proceed in a phased approach. This approach necessitates the further segmentation of the selected VLANs into sub-VLANs for testing to minimize the scope of the affected systems. This ensures that business continuity is maintained in case the sub-VLANs for testing require an extended period to configure.

As part of the testing process, it is crucial to determine if the removal of any component of the cipher suite would adversely impact the current servers. To achieve this objective, configurations will be done on [Redacted] will be utilized to simulate servers and clients with identical settings to the servers being tested. The IIS Crypto tool can be employed to disable weak ciphers in the testing environments, and the consequences carefully monitored. However, this process is time-consuming, as it involves disabling individual ciphers one by one and conducting periodic tests. Once an optimal combination of the cipher suite has been established, a GPO for that cipher suite will be created, and ready for implementation on the actual servers.

### 3. Deploying

After concluding the testing phase, the GPO template resulting from the testing will be implemented to effect the required changes on servers that contain weak cipher suites. To prevent server downtime, automation tools will not be utilized at this stage. The update will be carried out during nighttime when server access is at its lowest point. This approach will provide enough time to respond to and resolve any issues that may emerge during the system upgrade.

Aside from GPOs, the device image with the current configurations and security hardening should be properly setup, to preserve all the relevant configurations that were done during the testing to ensure that once the device has been deployed, the environment that it is in is preserved as well. This establishes a seamless deployment from the testing environment to the production environment.

### 4. Monitoring

During the final stage of the upgrade, appropriate monitoring tools may be employed to keep track of changes made to the servers. These monitoring tools can send log files to a central management system like SIEM, providing a comprehensive view of the servers' performance. This will enable the team to determine if the application of the new cipher suite will cause any decrease in server performance and identify any connection issues between servers and clients. As a result, alternative cipher suites or other methods to address encryption weaknesses can be considered.

### Summary of Year-1 Recommendations

To provide a summary of the steps for year 1, the objective is to enhance the existing state of the ciphers implemented in Bow Valley College. This will be achieved through the implementation of [Redacted].

However, selecting which encryption or cipher suits to implement is only the initial stage. The subsequent phase, which will require considerable time, will be dedicated to testing and ensuring that the systems are compatible with the proposed changes. The testing process will entail identifying the VLANs to prioritize, which will depend on several factors, including the number of operational systems, their criticality, and the availability of their configurations.

The testing, deployment, and monitoring processes will be implemented through a phased approach. This approach will involve the further subdivision of VLANs to minimize the number of affected devices. The steps for testing, deployment, and monitoring will all be recursive until all the VLANs have been updated.

## Year-2: Anticipatory Planning for Post-Quantum Break

Now that the implementation of strong ciphers has been completed, the focus of the second year is on anticipatory planning for the transition to post-quantum cryptography implementation. It is worth noting that the NIST is currently in the process of standardization, with the 4th round still ongoing. The final selected standards are expected to be announced by 2024, which coincides with the 2nd year planning phase.

While there is no official standard yet, significant planning efforts are underway to consider the hybrid approach. This approach involves the integration of current strong ciphers with known quantum resilient ciphers. The primary reason behind the hybrid approach is compliance purposes as regulators such as Payment Card Industry Data Security Standard (PCI DSS) are also waiting for appropriate standards to be put in place for the post-quantum era. It is considered a viable option as it allows for the utilization of the existing strong ciphers while also providing protection against quantum attacks. The planning efforts for the second year are aimed at ensuring a seamless transition to post-quantum cryptography. The breakdown of the planning can be seen in Table 9 found below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Year | Start | End | Duration (Days) | Label |
| Second Year | **Planning/Preparing to Apply Quantum Safe** | | | |
| 5/1/2024 | 7/29/2024 | 90 | **Create/Update Policy and Standards for quantum requirements**  **May 1 - Jul 29** |
| 7/30/2024 | 10/27/2024 | 90 | **Prioritization of Systems for Quantum**  **Jul 30 - Oct 27** |
| 10/28/2024 | 2/24/2025 | 120 | **Selecting Quantum Candidate for Each System**  **Oct 28 - Feb 24** |
| 2/25/2025 | 4/25/2025 | 60 | **Plan for Transition**  **Feb 25 - Apr 25** |

Table 9: Breakdown of Schedule for Year-2

As recommended by the Department of Homeland Security, this is the guidance that the team will follow for the course of the year-2 planning: (Preparing for post-quantum cryptography, 2021)

First, we must apply NIST Policy and Standards for quantum resilient acquisition, cybersecurity, and data security standards that will need to be updated to suit needs.

Prioritization of Systems for Quantum could be based on prioritizing one system over another for Cryptographic transition organization functions, goals, and needs each activity a part in system prioritization for quantum. There are some questions to help us define that.

Is the system a high value asset based on organizational requirements?

* + What is the system protecting (e.g., key stores, passwords, root keys, singing keys, personally identifiable information, sensitive personally identifiable information)?
  + What other systems does the system communicate with?
  + To what extent does the system share information with other entities outside of your organization?
  + How long does the data need to be protected?
* For selecting what Quantum Candidate, we could use on each system, this step could be the longest step on our roadmap. Because we need to know how to apply the changes to the current system. To do that, we will create a testing environment and go through many tests.
* Based on what we get from the previous phase, we could use inventory, prioritization, and testing information to develop a plan for systems transitions upon publication of the new post-quantum cryptographic standard. Cryptographic agility should be included in transition plans to assist future modifications and provide flexibility in unexpected developments.

In the context of achieving quantum resiliency in cryptographic systems, several cybersecurity organizations are researching the integration of pre-quantum and post-quantum ciphers as a hybrid solution. This approach aims to provide protection against quantum attacks while also utilizing existing pre-quantum ciphers.

Although ongoing research indicates that there may be more developments in the hybrid approach by 2024, this paper focuses on introducing the four candidates for standardization. The purpose of this is to enable the institutions to plan and determine which ciphers are suitable for their specific needs and security requirements.

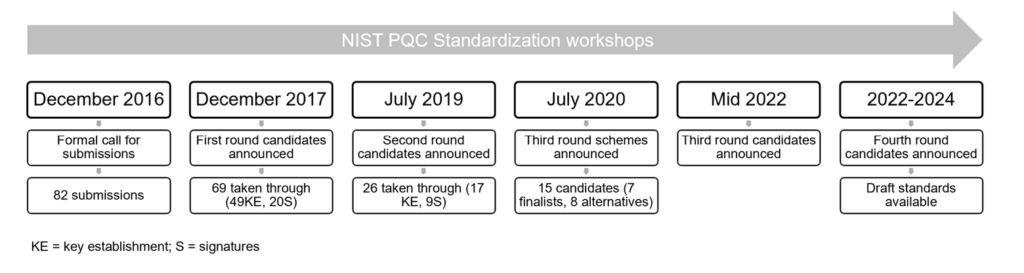


Figure 3: NIST timeline (The Quantum Insider)

### NIST Candidates for Standardization

The 4 candidates are as follows:

#### 1. CRYSTALS-KYBER

In 2017, NIST received the first round of submissions, and CRYSTALS-KYBER was one of the successful candidates. Modern computers use Public-key cryptography to ensure secure communication. But public-key cryptography security depends upon how difficult it is to solve certain mathematical problems. However, in comparison to classical computers quantum computers can solve some of these problems much faster, hence it makes public-key cryptographic algorithms vulnerable to attacks. As a result, there is a requirement for post-quantum resilient cryptographic algorithms that can provide security against quantum attacks.

It is a lattice-based encryption algorithm that provides security against quantum attacks. It is based on the difficulty of the learning with errors (LWE) problem, which is believed to be resistant to attacks by both classical and quantum computers.

##### Implementation of CRYSTALS-KYBER in Bow Valley College

For the implementation in Bow Valley College, the first thing that needs to be considered is to download the source code from the CRYSTALS-KYBER website. Then we are required to compile the code and produce the necessary libraries. The System requirements for running CRYSTALS-KYBER also need to be checked and confirmed.

Once we have set up the necessary environment for running CRYSTALS-KYBER, we can then start testing its performance. We will compare the performance of CRYSTALS-KYBER with traditional cryptographic algorithms such as RSA and ECC. Also, the measurement of the performance with reference to encryption and decryption time, key generation time, and key size will be made.

The college can implement CRYSTALS-KYBER by following the guidelines provided in the Round 1 to Round 4 Submissions from NIST. The first step would be to understand the requirements and specifications of the algorithm. It has a security level of 2^128 against quantum attacks and provides both confidentiality and authenticity.

Next, the college would need to choose appropriate parameters based on the security level required and the resources available. Different parameters are assigned using this algorithm, and the larger the parameter assigned and set, the higher the security level but also the more computational resources utilized. The parameters also need to be carefully chosen to ensure that they are not vulnerable to side-channel attacks.

After the selection of the parameters, the college can implement the CRYSTALS-KYBER algorithm. It consists of two main phases: the key generation phase and the encapsulation/decapsulation phase. In the key generation phase, by using LWE problem a public/private key pair is generated. During the phase of encapsulation/decapsulation, a shared secret key is encapsulated using the public key and then decapsulated using the private key.

To verify post-quantum resiliency, it is required by the college to stay up to date with the latest developments in post-quantum cryptography. This would include following NIST's Post-Quantum Cryptography Standardization project and staying informed about the latest submissions and recommendations. The college would also need to ensure that its implementation is secure against both classical and quantum attacks by following best practices in cryptographic implementation and testing.

After implementing CRYSTALS-KYBER in Bow Valley College, we will find that it performs better than traditional cryptographic algorithms such as RSA and ECC in terms of key size and encryption, and decryption time. The key generation time for CRYSTALS-KYBER is slightly slower than RSA and ECC. However, this is a one-time process and does not affect the overall performance of the system.

#### 2. CRYSTALS-DILITHIUM

Crystal-Dilithium is defined as the digital signature which is quantum-resistant algorithm, which is more secure version than the RSA and elliptic curve cryptography and can replace these ciphers easily.

Crystal-Dilithium is a quantum-safe algorithm that is based on a digital signature scheme that is strong when built upon module learning with errors. This means that any threat actor who has access will not be able to produce the key or the signature to decrypt the message.

Based on the video by International Association for Cryptologic Research (TheIACR, 2019), Dilithium is much faster in key generation than other algorithms and that is why it is easy to implement with no Gaussian sampling as compared to Falcon, that uses Gaussian sampling. Furthermore, the key size and signature for Crystal-Dilithium is small because signature compression is done which was based on two previous research back in 2012 and 2014. This algorithm is more valuable because of Conservative parameter selection giving more flexibility when cryptanalytics is being done.

**Implementation of CRYSTALS-DILITHIUM in Bow Valley College**

Given that NIST chose CRYSTALS-Dilithium Algorithm as their primary choice, it is in Bow Valley College best interest to try to implement this digital signature encryption first over FALCON and SPHINCS+. During the round 2 of the NIST project, the researchers of Crystals-Dilithium proposed a variant called Dilithium-AES *(Dilithium, n.d.)* which uses AES-256 that is one of the encryption algorithms deployed in the Bow Valley College infrastructure and is something that the college can focus their research on. If this were to be implemented in the institution, it is expected to bring a more reliable and secure environment to the network, especially since it can be used in a hybrid mode in combination with an established pre-quantum signature scheme as current research suggests. The Open Quantum Safe (OQS) project, which is an open-source project that is meant for developing and prototyping the integration of quantum-resistant cryptographies with pre-quantum cryptographies (*About Our Project*, n.d.), is a platform that can be utilized by Bow Valley College to try and experiment with, to see how Crystals-Dilithium can be implemented together with the current pre-quantum ciphers employed in the college. If Crystals-Dilithium is successfully implemented, it is said to achieve more than 128 bits of security against all known classical and quantum attacks. However, Bow Valley College would have to implement security and computational software to match the standardization criteria of NIST. (Dilithium. n.d.).

#### 3. FALCON

Falcon is one of the candidates of NIST that is to be standardized as a cryptographic signature algorithm, which was submitted in 2017 by a group of people namely Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang. Falcon stands for “**Fa**st Fourier **l**attice-based **co**mpact signatures over **N**TRU”. Falcon is a Signature scheme based on the GPV framework and relies on NTRU lattices. In the current stage, Falcon has two variants: Falcon-512 and Falcon-1024. (Falcon, 2023)

##### Implementation of FALCON in Bow Valley College

Presently, NIST has not yet announced the final standards for Post Quantum Cryptography (PQC), but Falcon is one of the currently selected digital signature algorithms undergoing evaluation that Bow Valley College can choose to implement, once it has been standardized. While waiting for the final announcements, developing a prototype to test the integration of Falcon using the OQS project. Moreover, other hybrid cryptography can be employed using the same OQS project can be used, in which quantum-safe public-key algorithms are utilized alongside standard public-key algorithms (such as RSA or elliptic curves) to ensure that the solution is no less secure than existing classical encryption.

#### 4. SPHICS+

SPHINCS+ is one of the four candidates in the NIST post-quantum resiliency and it is a stateless hashing digital signature scheme. The construct of SPHINCS+ is a combination of a tweakable hash function, WOTS+, Hypertree or XMSS, and FORS (Bernstein et al., 2019). The combination of tweaking a hash function, one-time signature scheme, three key generation scheme, and random one-time key generation can help protect against quantum attacks and traditional linear processing attacks. Currently SPHINCS+ has three signature schemes such as SPHINCS+ SHAKE256, SHA-256, and Haraka (SPHINCS+, n.d.).

##### Implementation of SPHINCS+ in Bow Valley College

Currently, as of 2023, it is advised by the Canadian government to wait for NIST and Cyber Centre to fully review the implementation standard of SPHINCS+ stateless signature hashing algorithm as it must also meet the standard requirements of Common criteria standard (Security, 2022). Once the SPHINCS+ has been thoroughly reviewed Bow Valley College can implement SPHINCS+ stateless hashing algorithm by removing or switching any currently weak hashing methods that are susceptible to weak attacks. Security professionals at Bow Valley College can also use SPHINCS+ for verifying keys and digital certificates/signatures (Housley et al., 2022). Lastly, Bow Valley College should remove the following that uses the traditional hashing methods for furthering quantum resilience.

Below are the methods that will become an issue if SPHINCS+ is not used in the environment:

* Public Key Infrastructure
* Digital signatures/certificates
* Two-factor authentication that uses the traditional method of creating digital certificates.

#### Other Post-Quantum Ciphers that can be considered

Below are other key exchanges that can be considered while waiting for the standards to be announced as they are said to be post-quantum resilient.

##### Hash-based cryptography, Lattice-based cryptography, or Code-based cryptography

These alternatives can withstand post-quantum computers and are resistant to quantum attacks, ensuring the security of communication channels even in the face of powerful quantum computers. They can help keep communication channels safe even if attack vectors have access to advanced quantum computers. These kinds of cryptography should be used over DH or RSA cryptographies which are known to be weak against a quantum computer.

##### Quantum Key Distribution (QKD)

QKD can be used as a replacement for ECDHE or PSK. QKD uses the principles of quantum mechanics to generate and distribute cryptographic keys. As a result, they are unaffected by all types of attacks, including quantum attacks. Thus, replacing ECDHE with QKD provides stronger quantum resilience and ensures secure communication channels.

##### Super Singular Isogeny Diffie-Hellman (SIDH)

SIDH is a key exchange algorithm that is based on the mathematical concept of isogenies and is resistant to all known classical and quantum attacks, ensuring that communication channels remain secure even in the presence of powerful quantum computers. To summarize, SIDH usage instead of ECDH will provide stronger quantum resilience, ensuring secure communication channels.

##### Post-Quantum Diffie-Hellman (PQDH)

The PQDH algorithm is designed for secure key exchange, even against quantum computer attacks. Its mathematical foundation is rooted in the complexity of isogeny graphs, which is thought to be beyond the reach of quantum computing capabilities.

##### New Hope

This key exchange algorithm relies on lattice cryptography. Its core technique involves the use of ring learning with errors (RLWE) to create a shared secret key. Experts consider New Hope to be resistant to quantum computer attacks, providing a promising solution for secure key exchange.

##### Other examples of post-quantum key exchange algorithms that can be looked at are:

* NTRUEncrypt
* FrodoKEM
* Supersingular Isogeny Key Encapsulation (SIKE)
* Code-Based Cryptography (e.g., McEliece)

### Summary of Year-2 Recommendations

To summarize, year-2 would be spent on a lot of planning, especially for the 4 currently selected NIST candidates. Soon there will be more available research regarding how the hybrid approach can be implemented and what configurations or upgrades are needed in the current systems. But for now, there are tools available such as the project called Open Quantum Safe (OQS) which is an environment where the hybrid approach of integrating pre-quantum cryptography with post-quantum cryptography can be utilized in creating prototypes and experiments to test how the integrated algorithms can be implemented in the institution. All the four candidates as well as some of the other post-quantum cryptographies such as NTRU-Prime, FrodoKEM, BIKE, Classic McEliece and HQC are available, for testing and prototyping in the OQS project platform *(Algorithms, n.d.).*

# Conclusion

In conclusion, the post-quantum break is inevitable, and all organizations would have to deploy some changes to their systems, specifically, the post-quantum ciphers that will be standardized. And to lay the groundwork for the Bow Valley College IT Security Team, the team has come up with two strategic objectives that serve as guidelines for the 2-year post-quantum break roadmap. Bow Valley College’s IT Systems have vulnerabilities with the presence of numerous weak cipher suites and because of that, the initial strategic objective is compliance and integration, which involves eliminating the usage of weak ciphers and ensuring a level of security that can withstand attacks from classical computers. Once the first strategic objective is achieved, it will leave Bow Valley College employing strong ciphers that conform to the current NIST standards and fortifies the system against potential cyber threats, ensuring compliance with regulatory standards and organizational security.

The second strategic objective is anticipatory planning for the transition to post-quantum cryptography. The hybrid approach—integrating current strong ciphers with known quantum-resilient ciphers—is considered a viable option for compliance purposes and protection against quantum attacks. The planning efforts for the second year are aimed at ensuring a seamless transition to post-quantum cryptography as standard solutions are still in the works.

Nevertheless, starting as early as now ensures that the research and especially the resources needed to implement the new standards would be well-planned and budgeted accordingly so that by the time NIST announces the final selected standards for post-quantum cryptography, Bow Valley College would be on track in implementing them as soon as they are available.

# References

*About our project*. (n.d.). Open Quantum Safe. <https://openquantumsafe.org/about/>

*Algorithms*. (n.d.). Open Quantum Safe. https://openquantumsafe.org/liboqs/algorithms/

Baeldung, & Baeldung. (2023). Hash vs. Message Authentication Code | Baeldung on Computer Science. *Baeldung in Computer Science*. <https://www.baeldung.com/cs/hash-vs-mac>

*Bar Mitzvah Attack.* (n.d.). Blackhat. <https://www.blackhat.com/docs/asia-15/materials/asia-15-Mantin-Bar-Mitzvah-Attack-Breaking-SSL-With-13-Year-Old-RC4-Weakness-wp.pdf>

Bernstein, D. J., Hülsing, A., Kölbl, S., Niederhagen, R., Rijneveld, J., & Schwabe, P. (2019). The SPHINCS + Signature Framework. *Computer and Communications Security*. <https://doi.org/10.1145/3319535.3363229>

c0D3M. (2021, December 12). Lucky 13 Attack Explained - c0D3M - Medium. *Medium*. <https://medium.com/@c0D3M/lucky-13-attack-explained-dd9a9fd42fa6>

*ChaCha20-Poly1305 and XChaCha20-Poly1305 — PyCryptodome 3.17.0 documentation*. (n.d.). <https://pycryptodome.readthedocs.io/en/latest/src/cipher/chacha20_poly1305.html>

CloudFlare. (n.d.). Why use TLS 1.3? *CloudFlare*. <https://www.cloudflare.com/en-ca/learning/ssl/why-use-tls-1.3/>

CodeSigningStore. (2022, April 7). *What Is the Most Secure Hashing Algorithm?* Code Signing Store. <https://codesigningstore.com/what-is-the-most-secure-hashing-algorithm>

*CVE - CVE-2020-5229*. (n.d.). <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-5229>

*CVE-2004-2761: The MD5 Message-Digest Algorithm is not collision resistant, which makes it easier for context-dependent attackers to co*. (n.d.). <https://www.cvedetails.com/cve/CVE-2004-2761/>

Dang, Q. (2008). *Recommendation for applications using approved hash algorithms*. Gaithersburg, MD, USA: US Department of Commerce, National Institute of Standards and Technology. <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-107r1.pdf>

Dilithium. (n.d.). <https://pq-crystals.org/dilithium/index.shtml>

*Fluhrer, S.R.,* Coppersmith, D. (1994, May 1). *The Data Encryption Standard (DES) and its strength against attacks*. Simonson. <https://simson.net/ref/1994/coppersmith94.pdf>

Gigamon, G. T. (2021). iTWire - What Is TLS 1.2 and why should we still care? *itwire.com*. <https://itwire.com/it-industry-news/strategy/what-is-tls-1-2-and-why-should-we-still-care.html>

Harwood, R., Yaport-Garcia, A., Henkel, I., Dknappettmsft, Downie, K., Parente, J., Coulter, D., Ross, E., Schonning, N., Hall, J., Short, M., & Andrei-Popov. (2023, February 14). *Transport Layer Security protocol*. Microsoft Learn. <https://learn.microsoft.com/en-us/windows-server/security/tls/transport-layer-security-protocol>

Housley, R. (2022, November 21). *Use of the SPHINCS+ Signature Algorithm in the Cryptographic Message Syntax (CMS)*. <https://www.ietf.org/id/draft-ietf-lamps-cms-sphincs-plus-01.html>

Junod, P. (n.d.). *Linear Cryptanalysis of DES* [Thesis]. Eidgenössische Technische Hochschule Zürich. <https://www.apprendre-en-ligne.net/crypto/bibliotheque/PDF/lincrypt.pdf>

Kelsey, J., Schneier, B., & Wagner, D. (1999). Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. In *Lecture Notes in Computer Science* (Vol. 1334, pp. 233–246). Schneier. <https://www.schneier.com/wp-content/uploads/2016/02/paper-relatedkey.pdf>

Kiprin, B. (2022, October 31). SSL BEAST Attack Explained | Crashtest Security. *Crashtest Security*. <https://crashtest-security.com/ssl-beast-attack-tls/>

*Knowledge Base*. (n.d.). Scanigma. <https://scanigma.com/knowledge-base>

Lake, J., & Lake, J. (2022). What is SHA-2 and how does it work? *Comparitech*. <https://www.comparitech.com/blog/information-security/what-is-sha-2-how-does-it-work/>

*Mantin, I., & Shamir, A. (2001). Weaknesses in the Key Scheduling Algorithm of RC4.* Cornell University. <https://www.cs.cornell.edu/people/egs/615/rc4_ksaproc.pdf>

*Microsoft security advisory: Update for disabling RC4 - Microsoft Support*. (n.d.). <https://support.microsoft.com/en-us/topic/microsoft-security-advisory-update-for-disabling-rc4-479fd6f0-c7b5-0671-975b-c45c3f2c0540>

*Microsoft security advisory: Update for disabling RC4 - Microsoft Support*. (n.d.). <https://support.microsoft.com/en-us/topic/microsoft-security-advisory-update-for-disabling-rc4-479fd6f0-c7b5-0671-975b-c45c3f2c0540>

Nir, Y., & Langley, A. J. (2015). ChaCha20 and Poly1305 for IETF Protocols. *ChaCha20 & Poly1305*, *2070–1721*. <https://doi.org/10.17487/rfc7539>

*NIST Retires SHA-1 Cryptographic Algorithm* | NIST. (2022). *NIST*. <https://www.nist.gov/news-events/news/2022/12/nist-retires-sha-1-cryptographic-algorithm>

*NVD - CVE-2005-4900*. (n.d.). <https://nvd.nist.gov/vuln/detail/cve-2005-4900>

*NVD - CVE-2009-3555*. (n.d.). <https://nvd.nist.gov/vuln/detail/CVE-2009-3555>

*NVD - CVE-2011-3389*. (n.d.). <https://nvd.nist.gov/vuln/detail/CVE-2011-3389>

*NVD - CVE-2013-0169*. (n.d.). <https://nvd.nist.gov/vuln/detail/cve-2013-0169>

*NVD - CVE-2013-2566*. (n.d.). <https://nvd.nist.gov/vuln/detail/CVE-2013-2566>

*NVD - CVE-2013-2566*. (n.d.). <https://nvd.nist.gov/vuln/detail/cve-2013-2566>

*NVD - CVE-2014-3566*. (n.d.). <https://nvd.nist.gov/vuln/detail/CVE-2014-3566>

*NVD - CVE-2016-2183*. (n.d.). <https://nvd.nist.gov/vuln/detail/CVE-2016-2183>

*NVD - CVE-2019-3730*. (n.d.). <https://nvd.nist.gov/vuln/detail/CVE-2019-3730>

*NVD - CVE-2021-26407*. (n.d.). <https://nvd.nist.gov/vuln/detail/CVE-2021-26407>

*OpenSSL-iana.mapping*. (n.d.). testssl. <https://testssl.sh/openssl-iana.mapping.html>

*Post-Quantum Cryptography | Homeland Security*. (n.d.). <https://www.dhs.gov/quantum>

Puneet. (2022). *What is SHA? What is SHA used for?* Encryption Consulting. <https://www.encryptionconsulting.com/education-center/what-is-sha/>

Security, C. C. F. C. (2022, July 6). *NIST announces post-quantum cryptography selections - Canadian Centre for Cyber Security*. Canadian Centre for Cyber Security. <https://www.cyber.gc.ca/en/news-events/nist-announces-post-quantum-cryptography-selections>

Soni, D., Basu, K., Nabeel, M., Aaraj, N., Manzano, M., & Karri, R. (2020). CRYSTALS-Dilithium. *Springer eBooks*, 13–30. <https://doi.org/10.1007/978-3-030-57682-0_2>

*SPHINCS+*. (n.d.). <https://sphincs.org/>

*SSL 3.0 Protocol Vulnerability and POODLE Attack | CISA*. (2014, October 17). Cybersecurity and Infrastructure Security Agency CISA. <https://www.cisa.gov/news-events/alerts/2014/10/17/ssl-30-protocol-vulnerability-and-poodle-attack>

*Supported Message Digest Algorithms*. (n.d.). <https://thalesdocs.com/gphsm/ptk/5.9.1/docs/Content/PTK-J/supported_MD.htm>

*Sweet32: Birthday attacks on 64-bit block ciphers in TLS and OpenVPN*. (n.d.). <https://sweet32.info/>

*TLS 1.2 vs TLS 1.1 - KeyCDN Support*. (n.d.). KeyCDN. <https://www.keycdn.com/support/tls-1-2-vs-tls-1-1>

*TLS Ciphersuite Search*. (n.d.) CipherSuite. <https://ciphersuite.info/>

TheIACR. (2019, June 17). *CRYSTALS-Dilithium: A Lattice Based Digital Signature Scheme* [Video]. YouTube. https://www.youtube.com/watch?v=UINdiUIp96Y

*What is Hashing Algorithm & How Does It Work? - AboutSSL*. (2020, August 10). About SSL. <https://aboutssl.org/what-is-hashing-algorithm-how-does-it-work/>

*What is MD5? Understanding Message-Digest Algorithms | Okta*. (n.d.). Okta, Inc. <https://www.okta.com/identity-101/md5/>