Published in the USA
Biogeosystem Technique
Issued since 2014.
E-ISSN: 2413-7316
2025. 12(1): 15-24

DOI: 10.13187/bgt.2025.1.15
https://bgt.cherkasgu.press

The Problem of Data Compromise in the Era of Quantum Computing

Rafek R. Abdullin [a], Maxim V. Dyuldin [b], Denis A. Egorov [a], Natalia V. Krupenina [a], Eugeny O. Olkhovik [a], Vladimir V. Karetnikov [a], Yuri K. Smirnov [a], Vasily Y. Rud [a, c, *], Daria A. Valiullina [d], Zhenyue Yuan [e], Van Yuikun [f], Alexey V. Cheremisin [b], Igor A. Chernov [a]

[a] Admiral Makarov State University of Maritime and Inland Shipping, Saint-Petersburg, Russian Federation
[b] Peter the Great St. Petersburg Polytechnic University, Saint-Petersburg, Russian Federation
[c] Ioffe Physico-Technical Institute, Saint-Petersburg, Russian Federation
[d] Kazan State Academy of Veterinary Medicine named after N.E. Bauman, Kazan, Russian Federation
[e] Shenyang Institute of Technology, Fushun, China
[f] Wenzhou University, Wenzhou, China

Paper Review Summary:
Received: 2025, June 18
Received in revised form: 2025, June 25
Acceptance: 2025, June 25

Abstract

Solving modern problems of ecology and the environment requires the use of large amounts of data and, accordingly, the use of quantum computers and artificial intelligence. This article examines the challenges associated with ensuring secure information exchange in the context of integrating quantum computers into communication networks. These quantum computers possess significantly higher processing speeds compared to standard von Neumann architecture computers. Considering that symmetric and asymmetric encryption algorithms rely on the assumption that breaking the used ciphers within a reasonable timeframe is practically impossible, the advent of quantum computers with their immense computational power calls into question the long-term stability of these ciphers. The primary threat posed by quantum computers is data compromise, which involves intercepting and accumulating encrypted data for future decryption attempts. To mitigate this threat, it is proposed to use a neural network to classify suspicious logs recorded by the operating system's security subsystem.

Keywords: quantum computing, post-quantum cryptography, cybersecurity, encryption algorithms, data compromise.

1. Introduction

Currently, there is rapid development in quantum technologies (Dushkin, 2018). With this progress comes the need to rethink existing approaches to data security. Classical encryption algorithms, both symmetric (e.g., AES) and asymmetric (e.g., RSA), have long served as the foundation for information protection. However, the emergence of quantum computers, capable of

* Corresponding author
E-mail addresses: ecobaltica@gmail.com (V.Y. Rud)

solving problems underlying these algorithms exponentially faster than classical computers, raises questions about their viability and resilience to new threats.

The main threats to standard encryption algorithms arising from quantum computers include attacks using Shor's algorithm, Grover's algorithm, and the compromise of long-term data (INFARS, 2025).

Symmetric encryption algorithms, such as DES and AES, use the same key for both encryption and decryption. While they are efficient and fast, their main drawback lies in the necessity of securely transmitting the shared key between sender and receiver. If the key is intercepted by an attacker, the entire security system becomes vulnerable.

An attack using Grover's algorithm, executed by a quantum computer, reduces the time required to brute-force keys by half, thereby weakening symmetric ciphers like AES. Although increasing key length can partially mitigate this vulnerability, it is insufficient in the long term (Burlakovs et al., 2020).

Most modern information systems utilize the asymmetric RSA encryption algorithm, which operates based on public-key cryptography. Its cryptographic strength relies on the difficulty of factoring large numbers, enabling a system where the public key is used for encryption and the private key for decryption. This innovation circumvented the problem of securely transmitting secret keys and revolutionized digital security.

The most cryptographically robust systems currently employ 1024-bit or larger numbers. However, increasing key size cannot be indefinite, as it prolongs encryption times and slows down data transmission.

Shor's algorithm is a quantum algorithm that performs factorization of large numbers in polynomial time, unlike classical methods, which require exponential time (Shor, 1997). This algorithm acts as a "cryptographic breaker." RSA relies on the complexity of this task for its security; thus, if quantum computers achieve sufficient power, they could easily derive private keys from public ones, rendering RSA entirely vulnerable.

The compromise of long-term data involves copying and storing intercepted data, even if quantum computers have not yet broken cryptographic algorithms. Attackers may already be collecting encrypted data today to decrypt it in the future when quantum technologies become widely accessible.

Considering all of the above, the most pressing issue in protecting information from quantum computer attacks is the problem of data compromise.

2. Objective of the Study

The primary problem under consideration is data compromise, i.e., situations where confidential information becomes publicly accessible due to leakage, interception, or unauthorized access. In the modern digital world, this can involve personal data (e.g., credit card passwords or passport details) as well as corporate, banking, or governmental information.

Compromise is particularly dangerous when it concerns long-term data. This means that even if data is encrypted using current algorithms at the time of transmission, its interception and subsequent storage by attackers could lead to potential exposure in the future, when numerous quantum computers surpassing classical von Neumann architecture machines in speed will operate globally. Encrypted records of bank transactions, medical records, or personal information protected today by RSA-2048 could become vulnerable in a few years due to advancements in quantum computing or improved cryptanalysis methods. Thus, while data may currently be protected, its long-term storage without regular updates to security mechanisms could result in serious issues.

3. Problem Statement

The most common and difficult-to-control factor is human error. Even the most complex and well-designed security systems are not immune to vulnerabilities like a password sticker attached to a monitor. Errors can also occur during implementation, as algorithms may be compromised due to implementation flaws or incorrect use of cryptographic libraries. This includes:

– Poor quality random number generation for key creation.
– Improper storage or transmission of cryptographic keys.
– Use of default security parameters that are easily predictable for attackers.

If credentials, passwords, and encryption keys are compromised, attackers gain immediate access

to previously encrypted information. The history of information security is replete with examples of master key breaches leading to extensive security violations. A notable example is the 2011 breach of DigiNotar, a Dutch certification authority, where hackers generated approximately 500 fake SSL certificates. This led to severe consequences, including a major man-in-the-middle attack on Gmail services, loss of DigiNotar's reputation, and eventual bankruptcy. Following this incident, Public Key Infrastructure (PKI) was reevaluated, and new protection mechanisms were implemented.

With the advent of new technologies like artificial intelligence (AI) and machine learning, the process of analyzing and extracting information has become significantly easier. Modern AI technologies enable faster and more accurate analysis of encrypted data, identifying patterns or minor deviations that can be exploited to attack cryptographic algorithms. Such systems can detect weaknesses in algorithm design, significantly increasing the risk of compromise. Additionally, many solutions rely on publicly available and widely used libraries, which can pose risks partly due to human factors. If a vulnerability is discovered in a popular cryptography library, it could jeopardize vast amounts of data protected by its tools.

3.1. Compromise identifiers (SIEM)

In MaxPatrol SIEM, there is functionality to check previously obtained events for the presence of indicators of compromise based on tabular lists. An indicator of compromise is a sign of suspicious activity or a malicious object within an organization's IT infrastructure; such signs may indicate an ongoing attack by adversaries. Indicators of compromise can include, for example, an IP address or domain name of a node where suspicious activity has been registered, or the hash value of a malicious file.

– To search for indicators of compromise in events, a task must be created to check events using one of the profiles of the "batcheventsearch" module. Standard profiles have been created for checking events based on data from reputational tabular lists:

– IOC search for IP addresses – for checking against IP addresses from the "repListIP" tabular list.

– IOC search for URLs – for checking against addresses or domain names from the "repListMasks" tabular list.

– IOC search for domain names – for checking against domain names from the "repListDomains" tabular list.

– IOC search for hash values – for checking against file checksum values from the "repListHashes" tabular list.

Based on these standard profiles, custom profiles can be created. Typically, a SIEM system is deployed over the protected information system and has an architecture of "data sources" → "data storage" "application server" (Figure 1).

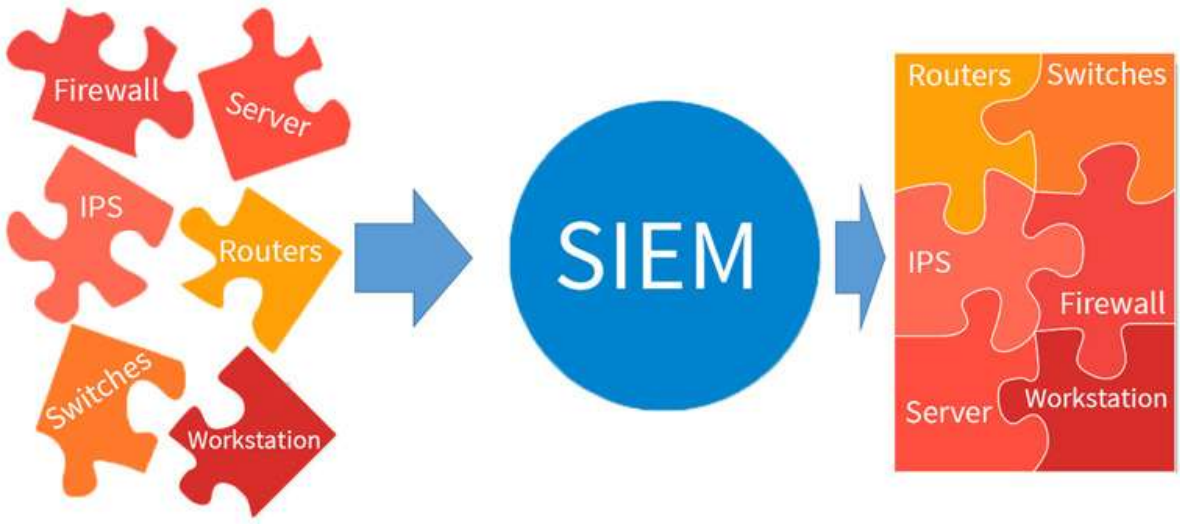

Fig. 1. SIEM-system

3.2. Indicators of Compromise (XDR)

Extended Detection and Response (XDR) is a multi-layered security technology that protects

IT infrastructure. This is achieved by collecting and correlating data from multiple security layers, including endpoints, applications, email, cloud environments, and networks, providing greater transparency into the organization's technological environment. This allows security teams to quickly and effectively detect, investigate, and respond to cyber threats.

XDR is considered a more advanced version of Endpoint Detection and Response (EDR). While EDR focuses on endpoints, XDR takes a broader approach, focusing on multiple security control points to detect threats faster through deep analytics and automation (Figure 2).

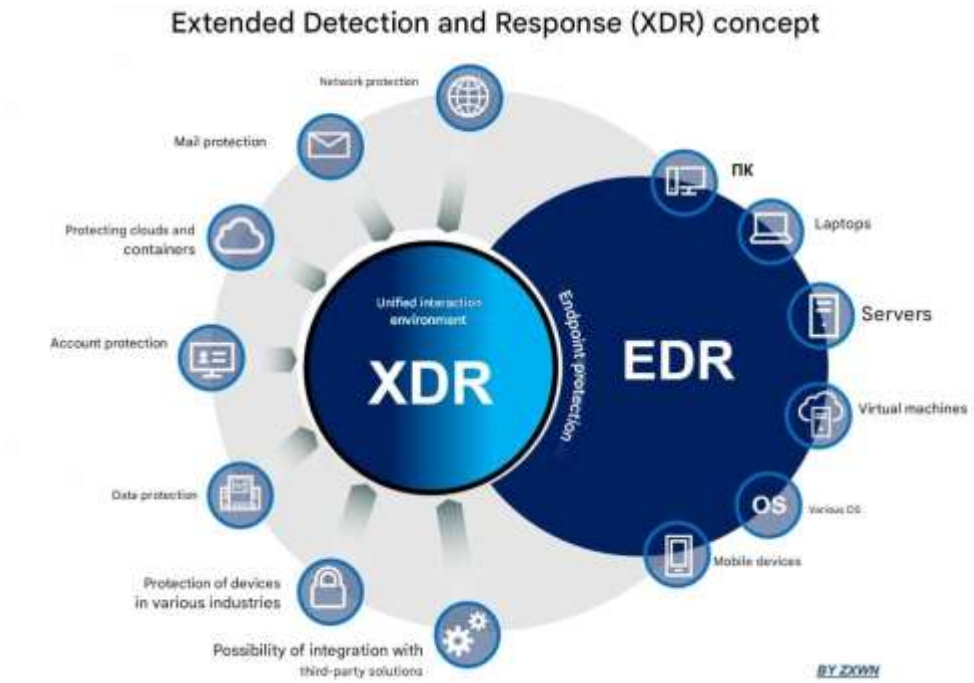

Fig. 2. XDR Conception

3.3. Key Indicators of Compromise

Key indicators of compromise serve as warning signs that an attack has been attempted, is ongoing, or has already occurred. Below are some of the main indicators:

3.1. Traffic Anomalies

Most organizations have a certain "profile" of normal network traffic. This profile reflects the expected behavior of users, applications, and services. For example, employees might typically transfer a specific amount of data during working hours. An indicator of compromise could be a sudden increase in outgoing traffic at night or connections to external IP addresses outside the trusted group. Such events may indicate data leakage or the presence of malware transmitting information to a control server.

3.2. Unusual Login Attempts

User work habits are generally predictable. They log in from specific geographic locations, during certain hours, and from the same devices. However, suspicious actions include:

– Attempting to log in at night if the user usually works during the day.

– Logging in from an IP address belonging to another country.

– Multiple failed login attempts for a single account, indicating a possible brute-force password attack.

3.3. Privileged Account Anomalies

Accounts with elevated privileges (e.g., administrator accounts) are prime targets for attackers because they provide access to sensitive data and critical systems. Signs that administrator rights are compromised include:

– Attempts to change access levels.

– Logins by administrators from unusual devices.

– Actions inconsistent with the user's usual responsibilities, such as mass deletion of data.

To protect privileged accounts, organizations implement Privileged Access Management (PAM) systems, multi-factor authentication, and action monitoring logs.

3.4. System Configuration Changes

Changes in system configurations may indicate the presence of malicious code, especially if:
– Security services or antivirus software have been disabled.
– Remote access has been enabled.
– Firewall settings have been altered.

Such actions are often performed during the early stages of an attack to prepare the infrastructure for exploitation.

3.5. Unexpected Software Installation or Updates

The sudden appearance of new software without IT approval may indicate an attempt at compromise. This type of software can appear due to employee negligence or phishing attacks. Examples of malicious software activity include
– The appearance of new executable files.
– Changes in startup processes.
– Installed services mimicking system services.

3.6. Multiple Requests for a Single File

If numerous accesses to the same file occur within a short time frame, this may indicate an attempt to bypass access restrictions or preparation for data theft.

To detect indicators of compromise, signs of digital attacks are recorded in log files. In cybersecurity practices utilizing IoCs, teams regularly monitor digital systems for suspicious activity. Modern SIEM and XDR solutions simplify this process using AI and machine learning algorithms, which establish baseline metrics for normal operations within an organization and then alert the team to anomalies. Additionally, it is important to educate employees who are not part of the security team, as they may receive suspicious emails or accidentally download infected files. Effective security training programs help employees better identify compromised emails and provide them with ways to report anomalies (Schöffel, 2021).

3.7. Solution Methods

Indicators of compromise (IoC) based on artificial intelligence (AI) allow for the detection of cyberattacks at early stages by analyzing not only static data but also behavioral patterns. Therefore, the problem of constructing a neural network for classifying threats based on Windows Security logs is highly relevant.

The illustration below demonstrates how the neural network will be trained based on data obtained from Windows Security logs, where the neural network learns from a CSV file derived from a security report.

The initial data for the program to classify threat logs is a CSV table containing the following fields (Figure 3. Part of the logs):
– Keywords,
– Date and time,
– Source,
– Event ID,
– Task category,
– Description/additional information.

Fig. 3. Part of the logs

Next, the data must be transformed for loading into the neural network:
– Removing rows with missing values,
– Processing CSV/text: accounting for comma-separated errors, merging columns,
– Encoding "EventID" values into classes 1, 2, and 3 based on severity.
Part of the code for label encoding and searching is shown in (Figure 4. Part of the code for labels).

```
# ============================
# 3. Assigning threat labels based on EventID
# ============================

# Example of mapping. Change it if necessary according to your requirements.
# Here:
# - 0: very dangerous
# - 1: you need to pay attention
# - 2: normal
mapping = {
    4624:2, # normal event
    4625: 0, # unsuccessful entry attempt is high-risk
    4648:1, # an attempt to log in with explicit authentication requires attention
    4672: 0, # privileged entry is very dangerous
    4720: 1, # account creation - requires attention
    4724:1, # password reset - requires attention
    4732: 1, # adding to a group requires attention
    4735:1, # group policy change - requires attention
    4738: 1 # Account change - requires attention
}

# Creating a new column with the threat label
df['ThreatLevel'] = df['Event_code'].map(mapping)
print("Example of threat labels:")
df[['Event_code', 'ThreatLevel']].shape
```

Fig. 4. Part of the code for labels

A neural network needs to be built for analysis and prediction based on network or personal computer logs to anticipate and predict attacks on devices or networks.
As the development environment, JupyterLab – an open-source integrated development environment – was chosen. The programming language and libraries used include:
– Python – a versatile programming language well-suited for big data analysis,
– Pandas – an analytical tool for big data analysis,
– Keras – a high-level library for building and training neural networks.
The choice of tool for the neural network was made based on an analysis of existing solutions for building neural networks (Table 1).

Table 1. Comparison of Libraries for Neural Networks

Name	Features	Language
TensorFlow	From Google, low-level, powerful, flexible. Used as a backend for Keras.	Python, C++
PyTorch	From Facebook, popular in academia, dynamic computation graph, flexible.	Python
Scikit-learn	Not for neural networks, but offers many ML models for classical tasks.	Python
MXNet	Supported by Amazon, cross-platform, scalable.	Python, R, Scala
JAX	From Google, optimized for acceleration and autodifferentiation.	Python

The Keras library was chosen due to its advantages, including:

– **Simplicity and ease of use**: Keras provides an intuitive API for rapid prototyping of neural networks, allowing developers and researchers to experiment with different model architectures easily.

– **Modularity**: Keras is organized into separate modules (layers, activation functions, loss functions, optimizers, etc.) that can be combined to create complex models, making the code more organized and maintainable.

– **Support for multiple backends**: Initially, Keras could operate on several computational engines (TensorFlow, Theano, CNTK). Currently, TensorFlow is the primary backend, ensuring broad support and integration with other tools in the TensorFlow ecosystem.

– **Large community and documentation**: Thanks to its extensive developer community and high-quality documentation, Keras has become a popular tool in both academic and industrial settings, facilitating the search for examples, templates, and support during model development.

– **Versatility**: With Keras, both simple models and complex deep learning architectures can be built for tasks such as classification, regression, image processing, text analysis, and more.

For building the neural network, the data is split into 80 % for training and 20 % for testing. The layers are then created as follows:

– **First Layer**: A fully connected (Dense) layer with 64 neurons and ReLU activation. The “input_dim” parameter specifies the number of features in the input vector. A Dropout layer with a probability of 0.3 randomly deactivates some neurons during training to prevent overfitting.

– **Second Layer**: A fully connected layer with 32 neurons and ReLU activation. Dropout is applied again.

– **Third Layer**: An additional layer with 16 neurons if a deeper network is required.

– **Output Layer**: Contains 3 neurons (corresponding to three threat classes). The softmax activation function converts output values into probabilities for each class.

The model is compiled with the Adam optimizer and the “sparse_categorical_crossentropy” loss function, suitable for multiclass classification tasks with integer labels.

Thus, a neural network has been constructed for training on prepared data to classify events by threat levels.

After all transformations, network training, and data classification, the built system distributes and outputs the results for convenient analysis by specialists. The program saves the obtained data in separate files, including:

– Training model graphs,

– Accuracy levels for each class,

– Overall model accuracy,

– Confusion matrix,

– Comparison with previous training iterations, showing how the neural network improved or changed.

On the graph below, we observe successful predictions by the neural network (Figure 5. Training Graph).

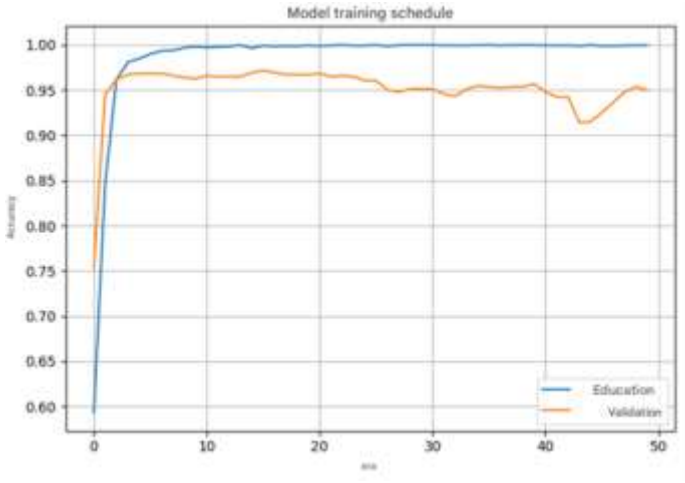

Fig. 5. Training Model Graph

In the screenshot below, we see accuracy on the test set, followed by precision, recall, F1-score, and the number of objects per class (Figure 6. Accuracy Table).

Точность на тестовой выборке: 0.9466

	precision	recall	f1-score	support
0	1.00	0.95	0.98	532
1	0.16	0.85	0.27	13
2	1.00	0.94	0.97	559
accuracy			0.95	1104
macro avg	0.72	0.91	0.74	1104
weighted avg	0.99	0.95	0.96	1104

Fig. 6. Accuracy Table

In Figure 7, the confusion matrix by classes is presented. For example, for class 0, 507 were correctly predicted, 25 were confused with class 1, and 0 were confused with class 2.

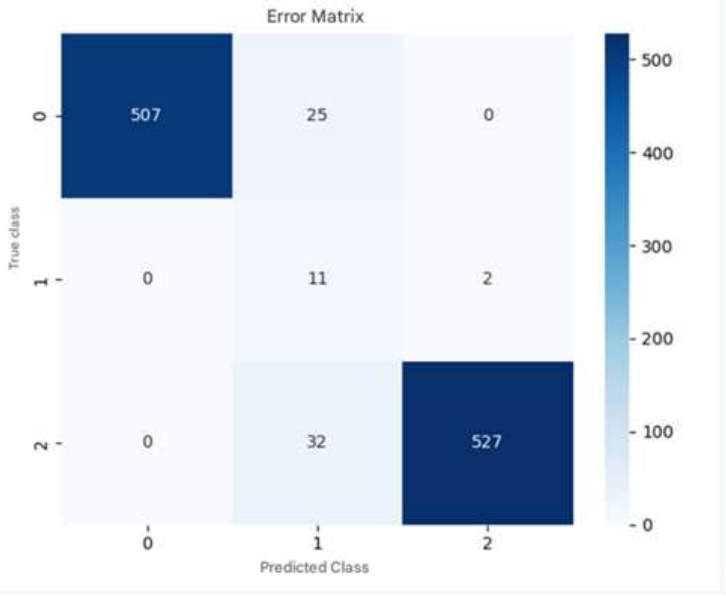

Fig. 7. Confusion Matrix

In Figure 8, a comparison with the previous model is shown. As seen, the new iteration did not improve compared to the previous model.

📊 Comparison with the previous model:
test accuracy: 0.4819 ↓ (-0.4647)
Test loss: 1.1260 ↓ (0.4641)
Accuracy: 0.5317 ↓ (-0.4565)
Completeness: 0.4819 ↓ (-0.4647)
f1 dimension: 0.4669 ↓ (-0.4971)

Fig. 8. Comparison with the Previous Model

4. Results

Thus, a neural network-based system for assessing and classifying security incidents based on Windows logs has been implemented programmatically. The system evaluates the potential

danger of each event and qualifies it into one of three classes with a probability of over 90 %. The remaining percentage represents potentially unclassified errors and threats, ensuring automation of analysis and threat detection. This system is a step toward building an automated protection system for computer networks and/or personal computers against attacks (Davydov et al., 2020; Li, 2017; Markley, 2014).

Through the creation of this program, the convenience of analyzing and predicting potential attacks on a network/computer has also been demonstrated.

To ensure security in the era of quantum computing (Hidary, 2019), quantum-resistant algorithms such as lattice-based algorithms (e.g., NTRU), Goppa codes, and isogeny-based elliptic curve systems must be used. These algorithms are considered sufficiently robust even against quantum computers (Hao, 2018). Additionally, further directions for implementing quantum-resistant algorithms are being developed, including:

– **Lattice-based cryptography**: Utilizes the complexity of finding the shortest vector in a lattice. Examples include NTRUEncrypt and Kyber.

– **Code-based cryptography**: Based on the difficulty of decoding random linear codes. A notable example is the McEliece algorithm.

– **Multivariate polynomial cryptography**: Relies on solving systems of nonlinear polynomials, such as Rainbow and GeMSS.

– **Quantum-secure cryptographic hash functions and protocols**, such as ring and supercode cryptography, offering new approaches to data protection.

These algorithms are in the process of standardization, and their implementation will help secure data in the post-quantum era.

5. Conclusion

Although quantum computers threaten many existing cryptographic systems, algorithms resistant to quantum attacks are continually being refined. These algorithms are being developed within the framework of post-quantum cryptography and are based on mathematical problems that cannot be efficiently solved even by quantum computers.

The analysis shows that while classical algorithms continue to provide the necessary level of security under von Neumann architecture, they face long-term threats from quantum attacks. This is particularly relevant for "harvest now, decrypt later" scenarios, where attackers copy encrypted information with the intention of decrypting it in the future. In this context, protecting long-term data becomes critically important.

The practical outcome of this work is the creation of a prototype system for intelligent analysis of Windows security logs using a neural network for threat classification. This demonstrates the potential of integrating machine learning methods into modern information security and data compromise monitoring systems.

Finalizing, the development of quantum technologies necessitates a reevaluation of existing approaches to information protection. The future of cryptography lies in the development of post-quantum algorithms, the implementation of hybrid cryptosystems, and the use of quantum cryptography. However, measures must already be taken today to protect long-term data by strengthening the implementation of cryptographic protocols, increasing key lengths, and introducing predictive threat analysis mechanisms.

References

Burlakovs et al., 2020 – *Burlakovs, J. Hogland, W., Vincevica-Gaile, Z., Kriipsalu, M.,Klavins, M., Jani, Y., Hendroko Setyobudi R., Bikse, J., Rud, V., Tamm, T.* (2020). Environmental Quality of Groundwater in Contaminated Areas—Challenges in Eastern Baltic Region. Water Resources Quality and Management in Baltic Sea Countries. Negm, A., Zelenakova, M., Kubiak-Wójcicka, K. (eds). Springer Water. Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-39701-2_4

Davydov et al., 2020 – *Davydov, R.V., Rud, V.Yu. and Yushkova, V.V.* (2020). On the possibility of analysis using the wavelet transform of the pulse waveform from the bloodstream. *J. Phys.: Conf. Ser.* 1695 012064 DOI: 10.1088/1742-6596/1695/1/012064

Dushkin, 2018 – *Dushkin, R.V.* (2018). Overview of the Current State of Quantum Technologies. *Computer Research and Modeling*. 10(2): 165-179. DOI: 10.20537/2076-7633-2018-10-2-165-179

Hao, 2018 – *Hao, Y., Xu, A.A.* (2018). Modified Extended Kalman Filter for a Two-Antenna GPS/INS Vehicular Navigation System. *Sensors*. 8: 3809.

Hidary, 2019 – *Hidary, J.D.* (2019). Quantum Computing: An Applied Approach. Springer International Publishing. Pp. 104-107.

INFARS, 2025 – INFARS. Post-Quantum Cryptography. 2025. [Electronic resource]. URL: https://infars.ru/blog/post-quantum-cryptography-kak-kvantovye-kompyutery-ugrozhayut-bezopasnosti-dannykh// (date of access: 03.05.2025).

Li, 2017 – *Li, T., Su, J., Liu, W., Corchado, J.M.* (2017). Approximate Gaussian conjugacy: Parametric recursive filtering under nonlinearity, multimodality, uncertainty, and constraint, and beyond. *Front. Inf. Technol. Electron. Eng*. 18: 1913-1939.

Markley, 2014 – *Markley, F.L., Crassidis, J.L.* (2014). Fundamentals of Spacecraft Attitude Determination and Control. Springer: New York, NY, USA.

Schöffel, 2021 – *Schöffel, M., Lauer, F., Rheinländer, C.C., Wehn, N.* (2021). On the Energy Costs of Post-Quantum KEMs in TLS-based Low-Power Secure IoT. Proceedings of the International Conference on Internet-of-Things Design and Implementation. Charlottesvle. VA, USA, 18–21 May. Pp. 158-168.

Shor, 1997 – *Shor, P.W.* (1997). Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput*. 26(5): 1484-1509.