УДК 519.7; 681.513.7; 612.8.001.57; 007.51/.52 DOI: 10.33744/2308-6645-2021-1-48-063-072
UDC 519.7; 681.513.7; 612.8.001.57; 007.51/.52

ОГЛЯД ПІДХОДІВ ОПТИМІЗАЦІЇ ОБМІНУ ДАНИМИ У МЕРЕЖІ МУЛЬТИСЕРВЕРНОЇ ІНФРАСТРУКТУРИ З ВИКОРИСТАННЯМ ШИФРУВАННЯ, КОМПРЕСУВАННЯ ТА ПРИСКОРЕНОЇ МАРШРУТИЗАЦІЇ

Гавриленко О.В., кандидат фізико-математичних наук, НТУУ «КПІ ім. Ігоря Сікорського», Київ, Україна, gelena1980@gmail.com, orcid.org/0000-0003-0413-6274

Шумейко О.А., Національний транспортний університет, Київ, Україна, shumeyko.ntu.edu.ua@gmail.com, orcid.org/0000-0003-2897-060X

Набоков Е.М., НТУУ «КПІ ім. Ігоря Сікорського», Київ, Україна, edwardnabokov@gmail.com, orcid.org/0000-0003-3528-3998

REVIEW FOR APPROACHES TO OPTIMIZE THE DATA EXCHANGE IN A NETWORK WITH MULTI SERVER INFRASTRUCTURE USING CRYPTOGRAPHY, COMPRESSION AND ACCELERATED ROUTING

Gavrilenko O., Candidate of physical and mathematical Sciences, NTUU «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, gelena1980@gmail.com, orcid.org/0000-0003-0413-6274

Shumeiko O., National Transport University, Kyiv, Ukraine, shumeyko.ntu.edu.ua@gmail.com, orcid.org/0000-0003-2897-060X

Nabokov E., NTUU «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, edwardnabokov@gmail.com, orcid.org/0000-0003-3528-3998

ОБЗОР ПОДХОДОВ ОПТИМИЗАЦИИ ОБМЕНА ДАННЫМИ В СЕТИ МУЛЬТИСЕРВЕРНОЙ ИНФРАСТРУКТУРЫ С ИСПОЛЬЗОВАНИЕМ ШИФРОВАНИЯ, КОМПРЕССИРОВАНИЯ И УСКОРЕННОЙ МАРШРУТИЗАЦИИ

Гавриленко Е.В., кандидат физико-математических наук, НТУУ «КПИ им. Игоря Сикорского «, Киев, Украина, gelena1980@gmail.com, orcid.org/0000-0003-0413-6274

Шумейко А.А., Национальный транспортный университет, Киев, Украина, shumeyko.ntu.edu.ua@gmail.com, orcid.org/0000-0003-2897-060X

Набоков Э.М., НТУУ «КПИ им. Игоря Сикорского», Киев, Украина, edwardnabokov@gmail.com, orcid.org/0000-0003-3528-3998

Introduction. Nowadays, the world utilizes the Internet for consuming and exchanging information. As for rule of thumb, any information is kept on servers. These servers are provided either by incorporated companies or public organization.

To retrieve any information from those servers, there is an existing solution that is used by all of the Internet users – BGP and DNS.

BGP is the compulsory component of the Internet. It allows to union different networks between each other. BGP solves the problem of routing between networks itself. Meanwhile, DNS is just a convenient wrapper, which allows people to avoid using IP address directly, but IP address instead. These 2 systems are fragile and monstrous. Nevertheless, it’s supported by both many public organizations and private companies. The end of the Internet era would come true only if BGP was broken. Until then we are good to use the Internet resources that it provides for us.

Any user requests information from one end being in another end. To support routing between any 2 ends, there are existing hardware computers: routers, hubs, switches. All of them are responsible for routing requests from end user to the target server and vice versa. In case of any computers knows nothing about target server, it sends request into outer network. As long as everything is determined in the computer networks, every target already exists in the world wide web.

Any request goes via hundred routers, where each of routers can read the content of the request. Here the problem comes. As long as we don’t control those routers, we cannot be sure that information is secure to send it in its original form. Therefore, it is invented cryptographical algorithm to encrypt request in a way, so only target server can decrypt it. It's called asymmetric cryptography or cryptography with public key. It means that anybody or anything can encrypt a request, but only target server can decrypt the request.

The main idea for cryptography is to change content of a request so it can be reverted back using public and private keys accordingly. Besides it's absolutely recommended to make all messages with the same length so to avoid giving any hints to the crackers [1]. So currently all encrypted messages look almost the same. Even though it makes data exchange more secure, it burdens the data exchange traffic in the world web. In other words, it sends 10Kb over a recipient in lieu of 1Kb or less. It works well for a single request but does not work for millions of requests. In addition, the problem of routing is still present. It means that any two requests might reach the same target using different routes. One of them can be longer, another one – shorter. How does routing impacts world traffic? Every request which goes via different routes is accumulated in those routers until it's sent further. A router is not black box, which sends requests impeccably and quickly. It has its own bounded resources like RAM and CPU. It can accumulate only certain number of requests, but not more. In case it cannot handle more requests, it denies all upcoming. It means the requests for google.com may not work at all. A router goes through the list of requests one by one, parses each one for source and target destinations and sends it further according to its route table. So, it's highly important to make all requests simple and small [2].

Moreover, there is routing problem. A router checks target destination of a request and tries to find the matching rule in its route table. If the one was found, it sends the request further. A router does not verify whether the target destination is sent with the shortest path or not. It picked up just the first matched one.

In this article is being reviewed different approaches to optimize data exchange in a network using cryptography – to make any date secure, compression – to make any data smaller and accelerated routing, which is based on meta information about target destination: how far they are, how loaded they are.

Formulation of the problem. There are 3 networks with 10 servers each. And there are 100,000 users who use the networks. Each user has an ability to save any encrypted information: music, movie, picture (later «object») in one of those servers and to request this information from the server. It's required to determine how to optimize (accelerate, simplified, reduce) data exchange.

Encryption of the data. Any information such as music, movie, picture is just a set of bytes, which consists of bits. Let's assume we have 1Kb of x_o that is required to be encrypted into x_e.

$$x_o(n_1, n_2, \dots, n_k) \to \; x_e(m_1, m_2, \dots, m_k) \qquad (1)$$

where x_o – original message, x_e – encrypted message, k – number of bits, n and m are 1 or 0.

It's a one-to-one mapping. Each bit is transformed into another bit by algorithmic rule. It means that the key with the size k can encrypt only $k\ - padding\ length$ of the original message. There are various key sizes: 128-bit (16 bytes), 256-bit (32 bytes), 512-bit (64 bytes), 1024-bit (128 bytes), 2048-bit (256 bytes), 4096-bit (512 bytes). Since we have 1Kb message, it's impossible to encrypt with any of those keys [3].

Therefore, there is the first approach. We have to split the original message at first. It's practical to split into parts, where 64 bytes each. It means we get 16 parts with 64 bytes each.

$$\begin{array}{c} x_o(n_1, n_2, \dots, n_{k1}) \to \; x_e(m_1, m_2, \dots, m_{k1}) \\ x_o(n_1, n_2, \dots, n_{k2}) \to \; x_e(m_1, m_2, \dots, m_{k2}) \\ \dots \\ x_o(n_1, n_2, \dots, n_{k16}) \to \; x_e(m_1, m_2, \dots, m_{k16}) \end{array} \qquad (2)$$

where x_o – original message, x_e – encrypted message, ki – number of bits, n and m are 1 or 0 for each part of the message. All k_i values are the same.

$$\Sigma_k\big(x_o(n_1, n_2, \dots, n_k)\big) \;\leftrightarrow\; \Sigma_k(x_e(m_1, m_2, \dots, m_k)) \qquad (3)$$

To encrypt 64 bytes of the message, we have to use key with the size bigger than 64 bytes. We may use 1024-bit key. As long as the key is bigger than a single part, there is a solution to solve it – padding. It allows you to add up dummy data into part to match the length of the key. Indeed, this dummy data adds up load to the Internet traffic. And it does not even provide any useful information [4].

Here is a rough calculation of the size of the encrypted message that is sent over the Internet from one user to another one:

The original message with the size of 1024 bytes is converted into 128 bytes of key * 16 (number of splits for original message). An equation looks like the following:

$$\begin{gathered}1024\ bytes \rightarrow \\ 128\ bytes\ of\ the\ key * 16 = 2048\ bytes\end{gathered} \tag{4}$$

As we see, the encrypted message weighs double bigger than the original one. It means that every encrypted message that is intended to be sent over the Internet will be double bigger. It loads the Internet traffic significantly [5].

To avoid that, there is a 2 approach. We have split message into such parts, so each one should be almost the same as a key size. It's worth to mention that it's not practical to encrypt 64 bytes with 512-bit key, because it's highly recommended to use padding to fix weaknesses of public key encryption. The padding may be fully randomized, which increases entropy of the original message. So, two exact the same messages can have different encrypted outcome. It does increase security facet [6]. Padding should be at least the size of 12 bytes to make it work. So, let's do backpropagation starting from key size of 512-bit (64 bytes).

$$\begin{gathered}64 - 12 = original\ message\ length\ (in\ bytes) \\ \frac{1024\ bytes}{52\ bytes} = 19 + padding\ for\ the\ last\ part\end{gathered} \tag{5}$$

So now it has the following equation (all in bytes):

$$\begin{gathered}1024\ original\ message \rightarrow \\ 64\ (key\ size) * 19 + 0.69 * 64 = 1260.30\ bytes\end{gathered}$$

The ratio of difference:

$$\frac{1260.30\ bytes}{2048\ bytes} * 100\% = 61.53\% \tag{6}$$

For now, we succeeded to diminish the resulting encrypted message in 61.53% of the previous result. It means that the traffic with encrypted data will be in 61.53% lighter [7]. It leads to accelerating the data exchange. Besides, routers/hubs/switches can afford to operate on more requests. Table is a comparing table between 2 approaches for cryptography:

Table 1 – Comparing table between 2 approaches for cryptography

Key size	Original message (bytes)	Encrypted message using 1 approach (bytes)	Encrypted message using 2 approach (bytes)	How effective the second approach is
128-bit (16 bytes)	1024	2048	4096	-100%
256-bit (32 bytes)	1024	2048	1638.4	80%
512-bit (64 bytes)	1024	2048	1260.31	61.5%
1024-bit (128 bytes)	1024	2048	1129.93	55.2%

There is an exceptional case when the key size is 128 bits. It's because the padding affects significantly due to being much bigger than the original message is. Nevertheless (table 2), we can see the tendency that the bigger file, the better 2 approach works.

Table 2 – Comparing table between 2 approaches for cryptography (resizing message)

Key size	Original message (bytes)	Encrypted message using 1 approach (bytes)	Encrypted message using 2 approach (bytes)	Ratio between first and second one
128-bit (16 bytes)	1024	2048	4096	-100%
256-bit (32 bytes)	10240	20480	16384	80%
512-bit (64 bytes)	102400	204800	126030	61.5%
1024-bit (128 bytes)	1024000	2048000	1129931	55.2%

As a conclusion for the encryption it's highly recommended to use padding for public key encryption. Splits of the any data must be flexible and be in size relative to the key size including padding. In addition, the bigger data, the less encrypted data we send over the Internet.

Compression of the data. Compression allows to shrink data and expand it back. It's responsible to convert one data form into another one. It works with 2 options:

- with loss
- without loss

$$x_o(n_1, n_2, \dots, n_k) \to x_c(m_1, m_2, \dots, m_z) \quad (7)$$

where x_o – original message, x_c – compressed message, k – number of bits of the original message, z – number of bits of the compressed message.

It's worth to note that the original message can be either encrypted or original one. The matter is about transferring less data via routers. Let's assume we use encrypted message as an original one. Its size is 1Kb.

So, the point of lossless compression algorithms is to transform data in a way it can be understood, read, decompressed by others meanwhile it weighs less than the original one. It means that compression works every time differently. It depends on characters that stand in a row, which are being swapped by compressing algorithm. Here is a vanilla example which illustrates how compression works:

$$AAAABBBBBBCCC \to A\backslash 3B\backslash 5C\backslash 2 \quad (8)$$

It's just a basic example, which shows algorithm, which is based on repeated elements. One problem that should be mentioned – the less data to compress, the more complicated to compress. The idea that comes around different existing algorithms is about learning past data to compress future data. Usually two third is compressed by algorithms. So, in case of 1KB data, we will obtain approximately 0.335KB. As long as we are trying to optimize the data exchange which goes inside the networks, a well determined and tested compressing algorithm can reduce a load of any network between users.

In addition, compression and encryption layers require to have opposite operation – decompression and decryption. Thus, it reduces load for networks to pass data, but it increases time to be able to read data that is being sent over the Internet.

Routing of the data. Besides implementing optimizations for encryption and decryption, there is one more valuable optimization. It's data routing in the networks. As far as we know, that any networks consist of a bunch of routers, hubs, switchers, it means that a path between any 2 vertices must be the shortest. It will allow to pass data without any additional operations of extra routers. [8]

It can be implemented in many different ways. As a starting point, we have to assume that currently routers just work based on their route tables. They find the matched pattern for request's destination in their tables and send it to another target router that is responsible for part of network. All routers, hubs, switches are discoverable using BGP protocol [9]. It unites all of them into a single big network. The internet would not live if BGP didn't exist. Table 3 is an example of routing table for any router [10].

Table 3 – Example of routing table for any router

#	Destination	Gateway
1	0.0.0.0/0	192.168.0.253
2	127.0.0.0/16	127.0.0.1
3	35.2.84.0/8	74.52.178.12
4	54.21.31.0/8	0.0.0.0
5	172.168.0.0/16	127.0.0.1

To optimize route-finding, we may rely on BGP. For now, routers work straightforwardly. To make it better, we have to implement a smart router. Its duties are searching, analyzing other routers and creating routing table that is based on how far routers are, how loaded they are. To do so, it requires to add up a new software layer, which must be highly optimized as well. It should preferably be a binary, which has the lowest footprint and works as fast as possible. This is important because it's a layer that adds up operational costs, and the time to send packet further is increased because of that.

Table 4 – Distance table

#	Destination	Gateway	Ping
1	0.0.0.0/0	192.168.0.253	100ms
2	0.0.0.0/0	192.168.0.255	90ms
3	127.0.0.0/16	127.0.0.1	1ms
4	35.2.84.0/8	74.52.178.12	54ms
5	54.21.31.0/8	0.0.0.0	10ms
6	172.168.0.0/16	127.0.0.1	2ms

The column with Ping results (table 4) shows us how far and how loaded the destination point is. For now, a router can redirect requests to low loaded routers. In the networks, distance which is physical is not as important as a ICMP response of the router from another end. Thus, we are good to use only ICMP requests like ping, to be about to determine how far and how loaded a router is. ICMP requests and its response can be assessed in ms. It represents a time within the requester gets its response. The bigger time, the more loaded or further a router is. Any routing problem can be illustrated as a graph. There are a lot of ready-to-go solutions, which solve different problems. One of the most popular and problematic tasks is a pathfinding. The edges of the graph can be represented with weights. Literally, weights are ICMP response time. It's worth to note that a router itself does ICMP requests to all its known endpoint to figure out how far and how loaded they are.

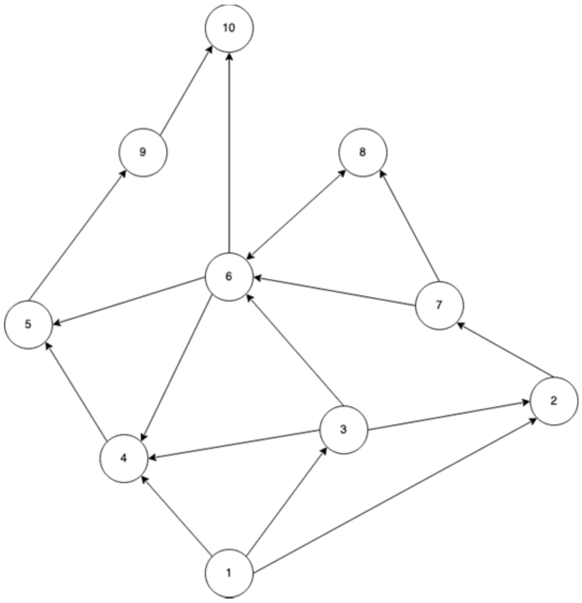

Figure 1 – Routing graph

It’s a hard problem to solve. Any router might lead to nowhere, so a packet will return to its own sender with an error of «Error destination». Routing itself works impeccably, it sends a packet to a subnet. There is a router for each subnet. The router must know where to send packet to. The shortest path from 1 to 10 will be the following:

$$1 \rightarrow 4 \rightarrow 5 \rightarrow 9 \rightarrow 10 \tag{9}$$

The total time to send data to a destination would take undefined time, because there is no awareness about current state of the network.

Although if there are weights for edges (connections between routers), then the outcome will be different.

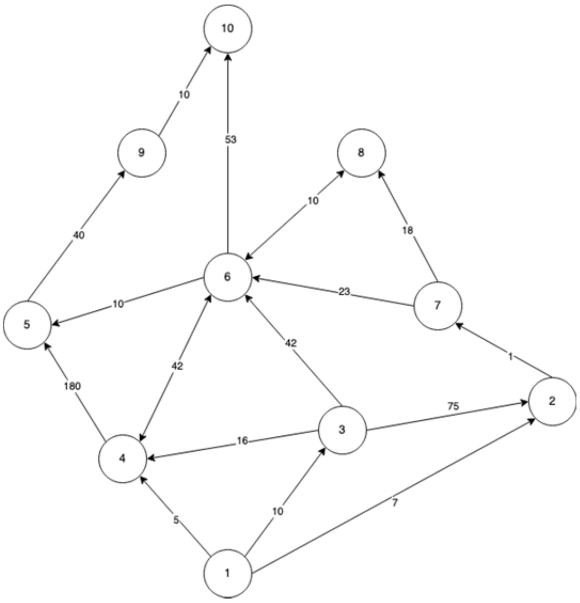

Figure 2 – Routing graph with latencies

After adding latencies between routers, the shortest path is now totally different:

$$1 \rightarrow 4 \rightarrow 6 \rightarrow 5 \rightarrow 9 \rightarrow 10 \tag{10}$$

The total time to send data would take 107 ms. For instance, the previous example would take 235 ms.

According to the greedy algorithm, we may find the shortest path if only we stick to the shortest latencies. This algorithm works well only when latencies is descending with the following router.

There are heuristic algorithms, which analyzes a few steps ahead. Literally, it makes recursively request to different routers and assess their assessment to their routers. It allows to accumulate more information and find out which router works the best to send request to.

Let’s assume we use this algorithm with 1 step ahead; the shortest path will be the following:

$$1 \rightarrow\ 2 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 9 \rightarrow 10 \tag{11}$$

With the heuristic algorithm, it would take only 91 ms.

As far as we see, it’s totally different from the results above. The only problem that to get information about other routers, it takes some time, so it’s highly dynamical approach to solve the shortest path in networks. The shortest path may go via more routers with the new approach in contrast to previous approaches. A path is based on latencies between routers, but not physical distance. It may have different quality of fiber channel between routers and so on. So, the system must be dynamical, but not statical, and rely on current state of their neighbors.

Table 5 – Comparison of the results of the algorithms

Plain algorithm	Greedy algorithm	Heuristic algorithm
235ms	107ms	91ms

As it's shown in the Table 5, it points that current routing in the networks is quite problematic and takes much more time than the one which is heuristic.

Conclusion. We reviewed different approaches how to optimize data exchange in networks. It helps significantly a lot when it all comes together. Encryption now consumes less memory and reduces load in networks. Compression shrinks data, which allows to transform 1KB data into tiny data, which is easier to operate with and to be sent to other recipients. In addition, sending data itself is a compulsory part of optimization as well. It was found out that the algorithms with heuristics work much better than greedy ones. It was assessed time within the packet is delivered to the destination.

In result of this analysis it was found out that current routing system in the Internet works worse than the results of proposals mentioned in this article.

Taking all these factors into account, it can be claimed that the Internet may work much faster and securer than it works currently in spite of the fact that decryption, decompression are expensive operations.

ПЕРЕЛІК ПОСИЛАНЬ

1. RSA algorithm [Електронний ресурс]. https://www.di-mgt.com.au/rsa_alg.html [in English].
2. RSA: Start validating your cybersecurity effectiveness. [Електронний ресурс]. https://www.verodin.com/post/rsa-2020-validating-cybersecurity-effectiveness [in English].
3. RSA key length [Electronic resource] https://www.javamex.com/tutorials/cryptography/rsa_key_length.shtml [in English].
4. How compression works for video materials [Електронний ресурс]. https://www.maketecheasier.com/how-video-compression-works/ [in English].
5. Lossy compression [Електронний ресурс]. https://en.wikipedia.org/wiki/Lossy_compression [in English].
6. Lossless compression [Electronic resource]. https://en.wikipedia.org/wiki/Lossless_compression [in English].
7. Difference between lossy and lossless compression [Електронний ресурс]. http://www.cvisiontech.com/resources/jbig2-compression-primer/lossless-lossy-perceptually-lossless-compression.html [in English].
8. A* path finding [Electronic resource]. http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html [in English].
9. A comprehensive study on pathfinding techniques for robotics and video games [Електронний ресурс] https://www.hindawi.com/journals/ijcgt/2015/736138/ [in English].
10. Border Gateway Protocol [Electronic resource] https://en.wikipedia.org/wiki/Border_Gateway_Protocol [in English].

REFERENCES

1. RSA algorithm [Electronic resource]. https://www.di-mgt.com.au/rsa_alg.html [in English].
2. RSA: Start validating your cybersecurity effectiveness. [Electronic resource]. https://www.verodin.com/post/rsa-2020-validating-cybersecurity-effectiveness [in English].
3. RSA key length [Electronic resource] https://www.javamex.com/tutorials/cryptography/rsa_key_length.shtml [in English].
4. How compression works for video materials [Electronic resource]. https://www.maketecheasier.com/how-video-compression-works/ [in English].

5. Lossy compression [Electronic resource]. https://en.wikipedia.org/wiki/Lossy_compression [in English].

6. Lossless compression [Electronic resource]. https://en.wikipedia.org/wiki/Lossless_compression [in English].

7. Difference between lossy and lossless compression [Electronic resource]. http://www.cvisiontech.com/resources/jbig2-compression-primer/lossless-lossy-perceptually-lossless-compression.html [in English].

8. A* path finding [Electronic resource]. http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html [in English].

9. A comprehensive study on pathfinding techniques for robotics and video games [Electronic resource] https://www.hindawi.com/journals/ijcgt/2015/736138/ [in English].

10.Border Gateway Protocol [Electronic resource] https://en.wikipedia.org/wiki/Border_Gateway_Protocol [in English].

РЕФЕРАТ

Гавриленко О.В. Огляд підходів оптимізації обміну даними у мережі мультисерверної інфраструктури з використанням шифрування, компресування та прискореної маршрутизації / О.В. Гавриленко, О.А. Шумейко, Є.М. Набоков // Вісник Національного транспортного університету. Серія «Технічні науки». Науково-технічний збірник. – К. : НТУ, 2021. – Вип. 1 (48).

У статті розглядаються різні підходи до оптимізації обміну даними в мережі з використанням криптографії для забезпечення безпеки даних, стиснення – для зменшення розміру трафіку даних і прискореної маршрутизації, заснованої на метаінформації про цільове пункті призначення, яка дозволяє оптимізувати маршрут пакетів даних між проміжними вузлами мережі.

Об'єкт дослідження – процес обміну даними в мережі.

Мета дослідження – оптимізувати процес обміну даними в мережі, для підвищення безпеки і швидкості передачі даних.

Методи дослідження – статистичний аналіз показників ефективності передачі даних.

При передачі даних, для забезпечення цілісності та безпеки застосовуються криптографічні алгоритми, ці алгоритми дозволяють обмінюватися даними за допомогою технології відкритого і закритого ключів. Але зараз цього може бути недостатньо, щоб ускладнити процес дешифрування даних, доцільно повідомлення, які передаються між вузлами, робити максимально схожими одне на одне, наприклад, однакової довжини. Таке рішення дозволить мінімізувати можливості розкриття шифру шляхом аналізу для зловмисника, але одночасно помітно збільшує обсяги переданих повідомлень за рахунок додаткового «маскувального» обсягу даних. Для компенсації цього ефекту в високонавантажених мережах пропонується застосувати алгоритм розумною маршрутизації, який дозволяє вибирати оптимальний маршрут слідування повідомлень з урахуванням завантаження маршрутизаторів і каналів між ними.

КЛЮЧОВІ СЛОВА: ЗАПИТ, ОБМІН ДАНИМИ, КОДУВАННЯ, МАРШРУТИЗАЦІЯ, КРИПТОГРАФІЯ

ABSTRACT

Gavrilenko O.V., Shumeiko O.A., Nabokov E.M. Review of approaches to optimizing data exchange in a multiserver infrastructure network using encryption, compression and accelerated routing. Visnyk National Transport University. Series «Technical sciences». Scientific and Technical Collection. – Kyiv: National Transport University, 2021. – Issue 1 (48).

In this article is being reviewed different approaches to optimize data exchange in a network using cryptography – to make any date secure, compression – to make any data smaller and accelerated routing, which is based on meta information about target destination: how far they are, how loaded they are.

The object of research is the process of data exchange in the network.

The purpose of the study is to optimize the process of data exchange in the network in order to improve the security and speed of data transfer.

Research methods – statistical analysis of data transmission efficiency indicators.

When transmitting data, cryptographic algorithms are used to ensure integrity and security, these algorithms allow the exchange of data using public and private key technology. But now this may not be enough to complicate the process of decrypting the data, it is advisable to make the messages transmitted between nodes as similar as possible to each other, for example, the same length. This solution will minimize the possibility of revealing the cipher by analysis for the attacker, but at the same time significantly increases the volume of transmitted messages due to the additional «masking» amount of data. To compensate for this effect in high-load networks, it is proposed to use a smart routing algorithm, which allows you to choose the optimal route for messages, taking into account the load of routers and channels between them.

KEY WORDS: REQUEST, DATA EXCHANGE, ENCODINGS, ROUTING, CRYPTOGRAPHY

РЕФЕРАТ

Гавриленко Е.В. Обзор подходов оптимизации обмена данными в сети мультисерверной инфраструктуры с использованием шифрования, компрессирования и ускоренной маршрутизации / Е.В. Гавриленко, А.А. Шумейко, Э.М. Набоков // Вестник Национального транспортного университета. Серия «Технические науки». Научно-технический сборник. – К.: НТУ, 2021. – Вып. 1 (48).

В статье рассматриваются различные подходы к оптимизации обмена данными в сети с использованием криптографии для обеспечения безопасности данных, сжатия – для уменьшения размера трафика данных и ускоренной маршрутизации, основанной на метаинформации о целевом пункте назначения, которая позволяет оптимизировать маршрут пакетов данных между промежуточными узлами сети.

Объект исследования – процесс обмена данными в сети.

Цель исследования – оптимизировать процесс обмена данными в сети, для повышения безопасности и скорости передачи данных.

Методы исследования – статистический анализ показателей эффективности передачи данных.

При передаче данных, для обеспечения целостности и безопасности применяются криптографические алгоритмы, эти алгоритмы позволяют обмениваться данными с помощью технологии открытого и закрытого ключей. Но в настоящий момент этого может быть недостаточно, чтобы усложнить процесс дешифровки данных, целесообразно передаваемые между узлами сообщения делать максимально похожими друг на друга, например, одинаковой длинны. Такое решение позволит минимизировать возможности вскрытия шифра путем анализа для злоумышленника, но одновременно заметно увеличивает объемы передаваемых сообщений за счет добавочного «маскировочного» объема данных. Для компенсации этого эффекта в высоконагруженных сетях предлагается применить алгоритм разумной маршрутизации, который позволяет выбирать оптимальный маршрут следования сообщений с учетом загрузки маршрутизаторов и каналов между ними.

КЛЮЧЕВЫЕ СЛОВА: ЗАПРОС, ОБМЕН ДАННЫМИ, КОДИРОВКА, МАРШРУТИЗАЦИЯ, КРИПТОГРАФИЯ

АВТОРИ:

Гавриленко Олена Валеріївна, кандидат фізико-математичних наук, доцент, доцент кафедри АСОІУ, ФІОТ, НТУУ «КПІ ім. Ігоря Сікорського», e-mail: gelena1980@gmail.com, тел. +380935768058, Україна, Київ, вул. Політехнічна, 41, 18 корпус, к. 430;

Шумейко Олексій Андрійович, доцент кафедри інформаційних систем і технологій, Національний транспортний університет, Київ, Україна, e-mail: shumeyko.ntu.edu.ua@gmail.com; тел. +38-044-280-70-66, м. Київ, М. Омеляновича-Павленка 1, к. 347а.

Набоков Едуард Максимович, НТУУ «КПІ ім. Ігоря Сікорського, e-mail: edwardnabokov@gmail.com., тел. +380934626303, Україна, Київ, вул. Політехнічна, 41, 18 корпус к. 430.

AUTHORS:

Gavrilenko Olena, Candidate of physical and mathematical Sciences, Associate Professor, Associate Professor of Department AIPSM, FICT, NTUU «Igor Sikorsky Kyiv Polytechnic Institute», e-mail: gelena1980@gmail.com, tel. +380935768058, Ukraine, Kyiv, Polytehnycheskayast., 41, 18 buld., r. 430;

Shumeiko Oleksii, National Transport University, Kyiv, Ukraine, e-mail: shumeyko.ntu.edu.ua@gmail.com, tel. +38-044-280-70-66, Ukraine, Kyiv, M. Omelianovycha-Pavlenka Str., 1, r. 347a

Nabokov Eduard, https://orcid.org/0000-0003-3528-3998, e-mail: edwardnabokov@gmail.com., tel. +380934626303, Ukraine, Kyiv, Polytehnycheskaya st., 41, 18 buld. 18, r. 430;

АВТОРЫ:

Гавриленко Елена Валерьевна, кандидат физико-математических наук, доцент, доцент кафедры АСОИУ, ФИВТ, НТУУ «КПИ им. Игоря Сикорского «, e-mail: gelena1980@gmail.com, тел. +380935768058, Украина, Киев, ул. Политехническая, 41, 18 корпус, к. 430;

Шумейко Алексей Андреевич, доцент кафедры информационных систем и технологий, Национальный транспортный университет, Киев, Украина, e-mail: shumeyko.ntu.edu.ua@gmail.com; тел. + 38-044-280-70-66., Г. Киев, М. Емельяновича-Павленко 1, к. 347а.

Набоков Эдуард Максимович, НТУУ «КПИ им. Игоря Сикорского, e-mail: edwardnabokov@gmail.com., Тел. +380934626303, Украина, Киев, ул. Политехническая, 41, 18 корпус к. 430.

РЕЦЕНЗЕНТИ:

Івохін Е.В. професор кафедри системного аналізу та теорії прийняття рішень факультету комп'ютерних наук та кібернетики КНУ імені Тараса Шевченка, доктор фізико-математичних наук, професор;

Аль-Амморі А.Н. завідуючий кафедри інформаційно-аналітичної діяльності та інформаційної безпеки Національного транспортного університету, доктор технічних наук, професор

REVIEWER:

Ivokhin E.V., Professor of the Department of Systems Analysis and Decision Theory, Faculty of Computer Science and Cybernetics, Taras Shevchenko National University, Doctor of Physical and Mathematical Sciences, Professor;

Al-Ammory A.N., Head of the Department of Information-Analytical Activity and Information Security of the National Transport University, Doctor of Technical Sciences, Professor