EasyChair Preprint

№ 5413

BORIS: Prototype of BGP Routing System Based on Blockchain Technology

Vitaly Antonenko, Fedor Sakharov and Daria Pluzhnikova

April 28, 2021

12nd IEEE International Conference on JointCloud Computing
23-26 August 2021; https://jointcloud.cloud/Home

BORIS: Prototype of BGP routing system based on blockchain technology

Antonenko Vitaly Aleksandrovich
Lomonosov Moscow State University
Moscow, Russia
anvial@lvk.cs.msu.su

Sakharov Fedor Vsevolodovich
Parity company
Moscow, Russia
fedor.sakharov@gmail.com

Pluzhnikova Daria Ruslanovna
Lomonosov Moscow State University
Moscow, Russia
pluzhnikovadari@mail.ru

***Abstract*—The BGP protocol, which is used for global routing, is not secure, because it does not have authentication mechanisms and authentication of the update source. In this regard, many solutions have been developed to ensure the security of BGP, but each of them depends on a central trusted node. The purpose of this work is to develop and implement a prototype of the dynamic routing protocol BGP based on blockchain technology, which allows you to build a secure decentralized system. The paper provides an experimental research of the prototype that shows that despite the increase in operating time, the prototype increases the security of the BGP protocol**

Keywords — blockchain, BGP routing, routing system, Internet

I. Introduction

Despite the fact that the Internet is now divided into subsystems, these components are nevertheless built on a centralized architecture. This leads to security issues, such as trust issues. For example, because the system is built on the trust of the system to the central node, it will be enough for an attacker to gain access to this node and then the attacker will own the entire system.

Now BGP is the main routing protocol on the Internet. The routing information is very important, that's why BGP requires TCP-like reliability. Once a TCP connection is established, network nodes start their communication by exchanging specific BGP messages to establish a session. After the BGP neighbors have moved to the stable state of ESTABLISHED, which means that the correct version of BGP is running and all settings are consistent, they proceed to exchange route information.

For this purpose, the UPDATE messages are used. Each UPDATE message can contain information about one new route or about deleting a group of old routes. Before saving the new path to the routing table, the latter passes through the filters defined by the AS routing policies.

BGP does not provide confidentiality, and BGP messages can be replayed. This means that an attacker can resend the UPDATE message, which causes an invalid route to be present in the router's BGP routing table. In addition, there is no mechanism that provides authentication of the origin of messages for BGP. In this case, attackers can impersonate one of the BGP nodes, so this approach can be used to introduce non-existent routes by using BGP message interception and analysis.

One of the main problems with BGP is that the path information in the received message must be trusted. Thus, the announcement of this information must be confirmed, for example, by an authoritative AS, that is, an AS that we can trust. However, BGP does not provide an authoritative hierarchy that allows a malicious or compromised AS to create and advertise and announce new non-existent or malicious routes.

In this regard, many BGP security solutions have been developed, such as SecureBGP (S-BGP), Secure Origin BGP (SoBGP), and Cross-Domain Route Verification (IRV) [1], [2], [3]. The disadvantage is that they each depend on a central, trusted node, which is enormous management overhead and complex.

II. Brief overview of existing solutions

There are many mechanisms for securing BGP. We will consider the most competitive protocols.

A. Secure-BGP (Secure Border Gateway Protocol)

Secure-BGP (Secure Border Gateway Protocol) Secure Border Gateway Protocol (S-BGP) is a modification of BGP designed to address security concerns. S-BGP uses digital signatures and X.509 certificates to create and validate BGP UPDATE messages advertised by AS. Secure-BGP is based on three different security mechanisms that seek to meet the security requirements of BGP. The S-BGP architecture uses security elements such as public key infrastructure (PKI), evidence, and Internet protocol security (IPsec).

S-BGP eliminates many of the security issues in BGP and provides security for the exchange of messages between ASs. However, it is susceptible to UPDATE message replay attacks. In addition, an attacker could remove signatures from messages, and further ASs would be unable to verify them.

One of the biggest problems with S-BGP is its complexity. This protocol is expensive to adopt and performance is degraded by the evidence and signature calculation for each UPDATE message. In addition, S-BGP has some storage requirements for route evidence, which makes it even more difficult to use.

S-BGP provides guarantees for secure BGP communications. Exchange paths and IP prefixes are verified and proven to be valid in destination autonomous systems using evidence. In addition, the integrity and confidentiality of updates is achieved through the use of the IPSec protocol. However, the high security of S-BGP requires a trade-off. Namely, its evidence storage requirements and performance degradation due to signature verification in every UPDATE message could discourage ISPs from deploying it to their ASs.

B. SoBGP (Sender Secure Border Gateway Protocol)

Secure Origin Border Gateway Protocol (soBGP) is a modification of BGP designed to address BGP security issues and improve communication reliability between peers. Like S-BGP, it uses PKI to authenticate and authorize objects on the network.

soBGP relies on several mechanisms, mainly certificates, to provide various security services for cross-domain routing. soBGP uses four types of certificates to verify peer prefixes and paths:

- Authentication certificate
- Authorization certificate
- Prefix Policy Certificate
- AS Policy Certificate

soBGP does not address all BGP security concerns. It has much weaker path authentication compared to S-BGP. The main difference between the approaches used by soBGP and S-BGP is that S-BGP provides dynamic path checking. This means that S-BGP speakers can view the topology and path of a message in real time. In contrast, soBGP uses databases that provide a static view of the topology and the paths within it. The UPDATE message received may have had a path from the changed topology that has not yet been reflected in the soBGP database. Topology changes can be accommodated by reapplying ASPolicyCert.

soBGP requires the database to be deployed on the AS side, certificate propagation can cause some deployment problems. The fact that there is an additional SECURITY message type can cause some difficulties in protocol deployment and backward compatibility.

soBGP provides a more flexible and lightweight solution than S-BGP for solving BGP security problems. The protocol can be tuned in such a way as to achieve a trade-off between security and cost. However, it does not provide path integrity, good source authentication, and an attacker can still invade the system. In addition, it has a weak protection mechanism for path authentication and PKI key distribution. Also, additional databases are required for path and policy validation, which complicates deployment.

C. IRV (Inter-Domain Route Verification)

Inter-Domain Route Validation (IRV) is the least centralized BGP security solution.

Each AS on the network contains an IRV server. Upon receipt of the UPDATE message, the BGP speaker will contact the IRV server in its AS to verify the correctness of the message received. The IRV server will verify the received request by requesting the IRV server from the AS referenced in AS_PATH. To check all ASs in AS_PATH, the IRV server will need to request information from all relevant IRV servers.

The main problem with IRV is that it must be able to connect to IRV servers in other ASs to check the path. This complicates the configuration and maintenance of the IRV server as it must communicate with other IRV servers.

In addition, deploying an IRV server requires a separate virtual machine or server that must be highly available and fault tolerant.

Unlike S-BGP and soBGP, IRV operations are independent of the routing protocol. Its security check is completely decoupled from BGP, allowing for more flexibility. However, there are some issues with deploying and maintaining the IRV

server that might render this solution suboptimal for real-world scenarios.

D. Conclusion

There are tradeoffs in the protocols reviewed that are not optimal for all scenarios. For example, S-BGP and IRV have strong security, but there are some performance and deployment issues. On the other hand, soBGP authentication is easier to deploy, but lacks some security features. As a result, there is still room for improvement and development of a better, more reliable and safer solution.

III. Development Software "Blockchain Substrate"

The prototype of the blockchain-based system is being developed focused on eliminating the problems that arise with the BGP security modifications described above. It shows how the blockchain can be used to validate messages and how to distribute information between nodes safely and securely by decentralizing the system.

This section discusses the main tools used in the implemented solution, such as the Substrate blockchain framework developed by Parity [6].

Blockchain is a technology that organizes a system consisting of a chain of blocks, each of which contains information about the previous ones. It is stored on all computers of the system participants at the same time, and the connection between the blocks is provided not only by numbering, but also by the fact that each block contains its own hash sum and the hash sum of the previous block. Storing copies of the blockchains on different computers independently makes it extremely difficult to change the information already included in the blocks, inasmuch as to change the information in one of the blocks, you will also have to edit all subsequent blocks.

You can also check the contents of the blocks, because each block contains information about the previous one. All the blocks are arranged in a single chain, which contains information about all the operations that have ever been performed in the system. Wherein, the created block will be accepted by other users if the numeric value of the header hash is equal to or less than a certain number.

Each block, in turn, consists of transactions. A transaction is considered complete and valid if its format and signatures are verified, and the transaction itself is grouped with several others and recorded in a block. There are two types of transactions: signed and unsigned.

Signed transactions contain the signature of the account that issued the transaction, and must pay a fee for including the transaction in the chain. Since the value of including signed transactions in the chain can be determined before they are executed, they can be transmitted in the network between nodes with a low risk of spam. In some cases, unsigned transactions can also be used, but the logic behind verifying them can be complex. Since the transaction is not signed, the commission is not paid. Because of this, there is no economic logic in the transaction queue to prevent spam. Unsigned transactions also lack a one-time number, making it difficult to protect against replay.

The client-side API of any Substrate-based blockchain allows a user to subscribe to a feed of events that take place in the runtime as the chain makes progress. Thus, Substrate notifies the user of a certain change in the state of the chain. The Substrate runtime module can generate events when it wants to notify external objects of changes or conditions in the runtime. The developer determines what events are generated in his module, what information is contained in these events, and when these events are generated.

IV. Quagga BGP Architecture

Quagga consists of several separate programs (daemons), each of which performs a specific function. For example, the bgpd daemon implements the BGP protocol. The main role in Quagga is played by the zebra demon. It receives route information from daemons that implement specific protocols and selects the best routes obtained from these sources. After that, these routes are passed to the Linux kernel, which transmits user traffic.

Each individual route in the routing table can be represented as a prefix with which different route information is associated. The Quagga routing table is simply a set of such prefixes with additional information associated with them. The zebra daemon stores all routes that have been passed to it or have been configured in zebra itself. Storing all routes allows you to quickly select a new best route if for some reason the current best route no longer exists.

When a new route is received, its routing information is added to the beginning of the linked list, after which all routes for

this prefix are sequentially scanned and the best one is selected. When a route is deleted, it is removed from the list of routes for this prefix, and the procedure for choosing the best route is started in the same way.

The BGP table is very similar in structure to the zebra routing table. Each prefix corresponds to several different routes received from different sources, but different BGP attributes are used instead of the administrative distance and metric.

For each route, a pointer is stored to the BGP neighbor from which the route was received, which allows you to use the corresponding data about the neighbor, for example, the connection type (IBGP or EBGP), its router id and IP address. When a new route is obtained or deleted, a procedure is started that selects the best one for this prefix by using a sequential pairwise comparison of routes.

First of all, when a new packet from a neighbour is received, it is parsed. Then the as path loop is checked, that is, it checks that the AS-PATH does not contain the autonomous system number to which the router belongs.

Then, various route filtering mechanisms are implemented if they are configured for the BGP neighbor from which the packet was received.

If the route has successfully passed all the filtering stages, then the incoming route-map is applied to it. Here you can flexibly modify the attributes of the received BGP route, or filter out the route. The last step is to ask zebra for the validity of the next-hop and the metric before it.

V. Overview of the Substrate-based solution

The blockchain provides the structure of a system that uses the blockchain to increase the security of BGP. Blockchain is a solution for an environment in which the parties do not have to trust each other, but must cooperate.

Advantages of using the blockchain:

- All transactions occur between nodes without any intermediary side. Moreover, there is no need for a third party for authentication.
- Provides immutability of announcement and the ability to re-track the BGP route chain. In addition, the route is checked by several parties and is more reliable.

Thus, the system has a common global blockchain. With each update, all AS in the group check to see if their path has changed with the update.

The functionality provided by the blockchain allows for a secure mechanism for storing and retrieving data in a publicly verifiable and immutable manner. For this reason, the blockchain provides a storage mechanism for collecting data and reaching consensus among participants.

BGP UPDATE messages can be verified using signed transactions running on the blockchain platform. When the router wants to send an Update message, it sends it as a signed transaction, thereby becoming a node of this blockchain network. This means that in order to interact with the blockchain and participate in the network, each AS router must run its own blockchain node.

After verifying the message and confirming the transaction, the blockchain makes changes to Storage, where the BGP routing table is stored, and interacts with other network nodes via Event messages. Now the BGP node does not need to send UPDATE messages to other nodes. Using the event module, the blockchain informs network nodes about changes in the routing table and the need for its local update.

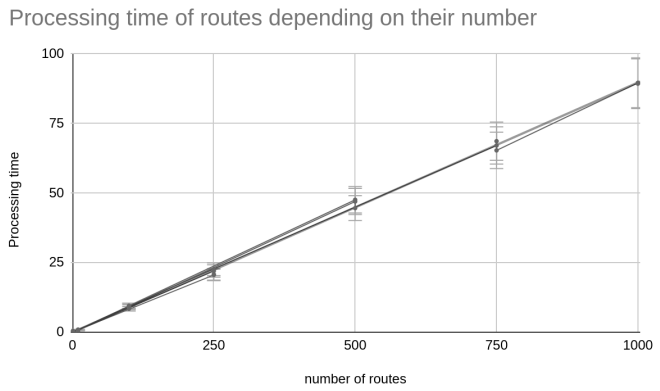
VI. Dicussion

The blockchain queues all received events and sends them to the block in the received order. Therefore, even if several events were received at the same time, the time of their total processing will be linearly dependent on the number of events received.

When testing the developed prototype of interaction without connected security systems, the following results were obtained:

TABLE I. System prototype test data

	Number of events (transmitted routes)			
	1	***10***	***100***	***1000***
Average working time (sec)	0.0807	0.8798	9.3947	89.487
Average processing time for one route (sec)	0.0807	0.0879	0.0939	0,0894



For comparison, the average transmission time of a single route in Quagga under the same conditions is 0.0002 s.

The increase in operating time is due to the TCP blockchain connection, as well as due to the parsing of routes and their inclusion in the blockchain.

VII. CONCLUSION

This article has reviewed the issue of BGP routing security. As its solution, the implementation of a prototype BGP routing system based on blockchain technology was proposed, which made it possible to ensure the security of transmitted routing data while maintaining a relatively high speed of information processing. The blockchain system, in contrast to the previously proposed solutions based on centralized systems, provides greater reliability of the system's functioning due to the abandoning of a main node, an attack on which can lead to the failure of the entire system.

Despite the fact that the prototype[11] is slower than the Quagga system, in the long term it provides greater security when transferring routes due to verification of the sender of transactions, decentralization of the system, invariability of route chain announcements and the possibility of re-tracking them.

Perhaps, in the future, this system, if it cannot completely replace the ordinary BGP routing, can become a good solution for modernizing systems where the security of route transmission plays a key role.

VIII. REFERENCES

[1] S. T. Kent, “Securing the border gateway protocol,”The Internet ProtocolJournal, vol. 6, no. 3, pp. 2–14, 2003.

[2] R. White, “Securing bgp through secure origin bgp (sobgp),”BusinessCommunications Review, vol. 33, no. 5, pp. 47–53, 2003.

[3] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. D. McDaniel, and A. D.Rubin, “Working around bgp: An incremental approach to improving security and accuracy in interdomain routing.” inNDSS, 2003.

[4] M. Saad, A. Anwar, A. Ahmad, H. Alasmary, M. Yuksel, and A. Mo-haisen, “Routechain: Towards blockchain-based secure and efficient bgp routing,” inIEEE ICBC, 2019, pp. 210–218.

[5] Q. Xing, B. Wang, and X. Wang, “Bgpcoin: Blockchain-based internet number resource authority and bgp security solution,”Symmetry, vol. 10,no. 9, p. 408, 2018

[6] P. Savola , “Backbone Infrastructure Attacks And Protections”, Technical Report, Internet Engineering Task Force,January 2007 .

[7] B. R. Greene and P. Smith. “BGPv4 Security Risk Assessment”,IS P Essentials Supplement, Cisco Press Publications, June 11th , 2002

[8] Tuna Vardar, "SECURITY INTERDOMAIN ROUTING", Helsinki University

[9] https://www.parity.io/substrate/

[10] http://www.nongnu.org/quagga/

[11] https://github.com/Pluzhnikovadari/Substrate