D. N. Kolegov, Yu. R. Khalniyazova, Threshold Diffie Hellman Protocol, Prikl. Diskr. Mat. Suppl., 2021, Issue 14, 79–81

DOI: 10.17223/2226308X/14/17

3. https://github.com/rurban/smhasher.

4. Hash function t1ha. https://github.com/PositiveTechnologies/t1ha.

UDC 004.056.55 DOI 10.17223/2226308X/14/17

THRESHOLD SCHEME OF THE HELLMAN DIFFIE PROTOCOL

D. N. Kolegov, Yu. R. Khalniyazova

A threshold scheme for the Hellman Diffie protocol on elliptic curves is proposed, which allows you to create and store the private key of a protocol participant in a distributed manner without the need to recover the key to perform cryptographic operations on this key.

Keywords: threshold cryptography, Diffie Hellman protocol, elliptic curves.

Let sk be the private key of a participant in the Diffie Hellman protocol. We will call the Diffie Hellman function the function DH(sk, Q), which takes as input the private key sk (scalar) and the point Q on the elliptic curve and returns the point sk Q. The Diffie Hellman protocol is usually understood as the following sequence calculations (G forming an element of a subgroup of prime order q of the group of points of an elliptic curve E over a finite field):

1) Alice generates a random number a $\in$ Zq, calculates the value A = DH(a, G) and sends it to Bob;

2) Bob generates a random number b $\in$ Zq, calculates the value B = DH(b, G) and sends it to Alice;

3) Alice calculates the shared secret as K = DH(a, B), and Bob as K = DH(b, A). The idea behind the Diffie Hellman threshold scheme is to generate

Give the private key of a protocol participant x using a distributed algorithm to some entities, and then delegate to them the calculation of the value of the function DH(x, Q) using threshold cryptography methods. We will call such entities agents, and a participant in the Diffie Hellman protocol will be a group of agents that represents one of the parties to the Diffie Hellman protocol and performs the steps established by the protocol.

If in the classic Diffie Hellman protocol a participant is an atomic entity (person, process, etc.), then now a protocol participant is a group of agent entities (people, processes, ...) that interact with each other through developed protocols in such a way that that to external entities (another protocol participant, outside observers, etc.) a group of agents is indistinguishable from an ordinary participant in the Diffie Hellman protocol. Moreover, based on the results of executing these protocols, any of the agents knows the public key of the group and can represent the group when interacting with another participant in the protocol. Since a group of agents is indistinguishable for external entities from an ordinary participant, for simplicity of presentation we will further assume that only one of the participants in the Diffie Hellman protocol is represented by a group of agents. In this case, it does not matter which particular protocol participant consists of agents, since the calculations performed by the participants are symmetrical.

The first stage of the proposed scheme is the generation of private key shares, as well as the calculation of the public key of the corresponding participant in the Diffie Hellman protocol. The calculated public key is known to each agent in the group,

and the corresponding private key is unknown to anyone and exists only in the form of generated shares. This stage is described by the key generation protocol.

The key generation protocol is based on the idea of distributed key generation, which is built using Feldman's verifiable secret sharing scheme [1]. Each agent Pj is assigned an index that determines the shares he receives in Feldman's scheme. For simplicity, we will assume that the index of agent Pj is the value j (any values on which the polynomials of the Feldman scheme are defined can be used as indices, if these values are different for all agents; they are set during the preparation phase, are not secret and must be known all agents). Then in any Feldman scheme the agent Pj receives a share f (j), where f is a polynomial of the Feldman scheme. In this case, the generation of the final shares on the agents occurs without the participation of the dealer, as a result of which the secret share of the private key of each agent is known only to himself. This is achieved due to the fact that each agent takes turns acting as a dealer in the Feldman scheme, sharing some random value ui, and then thanks to the property of the Feldman scheme (the sum of the shares of ai and bi from the values of a and b, respectively, is a share of the value of a + b) agents, adding the obtained values, receive new shares from the value x = ∑

i ui, which was not split explicitly.

During protocols, agents exchange public values with each other, using commitment schemes to record the values exchanged. The notation used to describe the protocols is taken from [2]:

1) Each agent Pi selects a random value ui ∈R Zq and calculates [KGCi, KGDi] = Com(yi), where yi = ui·G; Com obligation scheme algorithm; KGCi algorithm-calculated liability; KGDi is a string that allows it to be expanded. The agent then sends KGCi to all agents.

2) Each agent Pi sends KGDi to all agents and then shares the value ui among all agents using the Feldman (t, n)-scheme. The public key of the corresponding participant in the Diffie Hellman protocol is equal to y = ∑ i yi = = ∑ i ui · G.

3) Each agent adds the shares obtained in Feldman's schemes to calculate its closed share xi. The final value of the private share of the agent xi is the share of the private key x = ∑ i ui in the Shamir (t, n)-scheme. At the same time the private key x is not restored at any time during the execution of the protocol and exists only in the form of shares.

To calculate the shared secret, a protocol for generating a shared secret is proposed. Receiving the public key Y of another participant in the Diffie Hellman protocol as input, agents calculate the shared secret S = x ∗ Y using their shares of the private key, without revealing them during the protocol:

1) Each agent Pi calculates the Lagrange coefficient λi and the value wi = λixi, which is the share of this agent in the additive (t – 1, t)-scheme for sharing the secret from the value x. Here λi is the value of the i-th basis polynomial in the Lagrange interpolation scheme at point 0.

2) The agent calculates the values Si = wi · Y , [EXCi, EXDi] = Com(Si), where Com is the obligation scheme algorithm, EXCi is the obligation calculated using the algorithm, and EXDi is a string that allows it to be expanded. The agent then sends EXCi to other participants.

3) Agents send each other EXDi values. The shared secret is S = ∑ i And.

The threshold scheme allows you to expand the number of scenarios for using the Diffie Hellman protocol, including the possibility of improving the security properties of the private keys of participants in the Diffie Hellman protocol: if in the classical scheme an attacker only needs to gain access to the network node that stores the corresponding private key, now the required number of nodes depends on the threshold and is always greater than one.

The proposed scheme is implemented for Diffie Hellman protocols on Curve25519 and NIST P-256 curves.

A preprint of the article is available on arxiv.org [3].

LITERATURE

1. Feldman P. A Practical Scheme for Non-interactive Verifiable Secret Sharing. http://www.cs.umd.edu/~gasarch/TOPICS/secretsharing/feldmanVSS.pdf.

2. Gennaro R. and Goldfeder S. Fast Multiparty Threshold ECDSA with Fast Trustless Setup. https://eprint.iacr.org/2019/114.

3. Kolegov D., Khalniyazova Yu., and Varlakov D. Towards Threshold Key Exchange Protocols. https://arxiv.org/abs/2101.00084.

UDC 003.26 + 004.056 DOI 10.17223/2226308X/14/18

USE OF RUSSIAN CRYPTOGRAPHIC ALGORITHMS IN THE NETWORK LEVEL SECURITY PROTOCOL WireGuard1

D. N. Kolegov, Yu. R. Khalniyazova

We consider the WireGuard cryptographic protocol, designed to ensure the security of the TCP/IP network layer and built using Russian cryptographic algorithms. The need to develop such a protocol is caused by the need to use WireGuard in domestic promising distributed and cloud technologies built using cryptographic information protection tools. The solution to this problem is described: the choice of primitives, their implementation, alternative approaches, aspects of software implementation and testing, the main current results of the work, as well as current areas of research. The developed specification and reference implementation can be used as a starting point for developing recommendations for standardizing the WireGuard protocol with Russian cryptographic algorithms.

Key words: WireGuard, GOST, VPN.

Currently, in the field of secure network technologies, protocols from the Noise Protocol Framework family are being actively researched, developed and applied. The main protocol here is WireGuard, which has recently also been supported in the Linux kernel.

WireGuard is freely available and open source software designed to replace the legacy IPsec protocol and its implementations. The undoubted advantages of WireGuard compared to similar

1The work was carried out by LLC ¾Safe Information Zone¿ and NPO ¾Kryptonite¿ as part of a joint research project.