Protokol vymeny klucov Diffie-Hellman Pseudokod

Algorithm 1 Bezpecna vymena klucov: Inicializacia relacie s docasnymi klucmi

function INITIALIZE_SECURE_SESSION(*master_key*, *is_initiator*)
    Vstup: *master_key* (32B tajomstvo), *is_initiator* (boolean)
    Vystup: *session_key* (32B) alebo kod chyby
    Obe strany:
    ephemeral_secret ← generate_random_bytes(32) ▷ Vygenerovanie tajneho X25519 kluca
    ephemeral_public ← crypto_x25519_public_key(*ephemeral_secret*) ▷ Odvodenie verejneho kluca
    if *is_initiator* **then** ▷ Strana klienta
        Cakat na verejny kluc servera
        peer_public ← recv_all(*socket*, 32)
        Poslat verejny kluc klienta
        send_all(*socket*, *ephemeral_public*, 32)
        session_nonce ← generate_random_bytes(24) ▷ Vygenerovanie nonce pre relaciu
        send_all(*socket*, *session_nonce*, 24) ▷ Poslanie nonce serveru
    else ▷ Strana servera
        Poslat verejny kluc servera
        send_all(*socket*, *ephemeral_public*, 32)
        Cakat na verejny kluc klienta
        peer_public ← recv_all(*socket*, 32)
        Cakat na nonce relacie
        session_nonce ← recv_all(*socket*, 24)
    end if
    Obe strany:
    shared_secret ← crypto_x25519(*ephemeral_secret*, *peer_public*) ▷ Vypocitanie DH tajomstva
    session_key ← setup_session(*master_key*, *shared_secret*, *session_nonce*)
    secure_wipe(*ephemeral_secret*) ▷ Bezpecne vymazanie tajneho kluca
    secure_wipe(*shared_secret*) ▷ Bezpecne vymazanie zdielaneho tajomstva
    return *session_key*
end function

Algorithm 2 Odvodzovanie klucov relacie

function SETUP_SESSION($master\_key, shared\_key, session\_nonce$)
    Vstup: $master\_key$ (32B), $shared\_key$ (32B z X25519), $session\_nonce$ (24B)
    Vystup: $session\_key$ (32B)
    $ctx \leftarrow$ crypto_blake2b_init(32) ▷ Inicializacia BLAKE2b kontextu s vystupom 32B
    crypto_blake2b_update($ctx, master\_key, 32$) ▷ Pridanie hlavneho kluca do hashu
    crypto_blake2b_update($ctx, shared\_key, 32$) ▷ Pridanie zdielaneho tajomstva do hashu
    crypto_blake2b_update($ctx, session\_nonce, 24$) ▷ Pridanie nonce do hashu
    $session\_key \leftarrow$ crypto_blake2b_final(ctx) ▷ Dokoncenie hashu do kluca relacie
    crypto_wipe(ctx) ▷ Bezpecne vymazanie hash kontextu
    return $session\_key$
end function

Algorithm 3 Protokol rotacie a validacie klucov

function ROTATE_SESSION_KEY($current\_key, block\_count$)
    Vstup: $current\_key$ (32B), $block\_count$ (pocitadlo)
    Vystup: new_key (32B) alebo kod chyby
    if $block\_count$ mod $KEY\_ROTATION\_BLOCKS = 0$ **then** ▷ Cas na rotaciu kluca
        Strana klienta:
        send_chunk_size($socket, KEY\_ROTATION\_MARKER$) ▷ Signal rotacie kluca
        Cakat na potvrdenie servera
        $rotation\_nonce \leftarrow$ generate_random_bytes(24) ▷ Vygenerovanie nonce pre rotaciu
        send_all($socket, rotation\_nonce, 24$) ▷ Poslanie nonce serveru
        Obe strany:
        $previous\_key \leftarrow$ copy($current\_key$) ▷ Ulozenie aktualneho kluca
        $current\_key \leftarrow$ rotate_key($previous\_key, rotation\_nonce$) ▷ Vygenerovanie noveho kluca
        Strana klienta:
        send_chunk_size($socket, KEY\_ROTATION\_VALIDATE$) ▷ Signal validacie
        $validation \leftarrow$ generate_key_validation($current\_key$) ▷ Vytvorenie validacneho kodu
        send_all($socket, validation, 16$) ▷ Poslanie validacneho kodu
        Strana servera:
        $client\_validation \leftarrow$ recv_all($socket, 16$) ▷ Prijatie validacneho kodu
        $our\_validation \leftarrow$ generate_key_validation($current\_key$)
        if $client\_validation \neq our\_validation$ **then**
            return ERROR_KEY_VALIDATION_FAILED
        end if
        send_chunk_size($socket, KEY\_ROTATION\_READY$) ▷ Signal pripravenosti
        secure_wipe($previous\_key$) ▷ Bezpecne vymazanie stareho kluca
    end if
    return $current\_key$
end function
function GENERATE_KEY_VALIDATION(key)
    Vstup: key (32B)
    Vystup: $validation$ (16B)
    $validation \leftarrow$ crypto_blake2b(key, "VALIDATION", 16)
    return $validation$
end function
function ROTATE_KEY($previous\_key, rotation\_nonce$)
    Vstup: $previous\_key$ (32B), $rotation\_nonce$ (24B)
    Vystup: new_key (32B)
    $new\_key \leftarrow$ crypto_blake2b($previous\_key \parallel rotation\_nonce \parallel$ "ROTATE", 32)
    return new_key
end function