Fudo ShareAccess: Security

Contents

Fudo Security Sp. z o. o.
Al. Jerozolimskie 178
02—486 Warszawa, Poland
NIP 52425217
sales@fudosecurity.com1

Introduction

Fudo ShareAccess is a cloud-based secure access management solution enabling remote access to enterprise resources through a zero-knowledge architecture. This document details the security mechanisms across three architectural layers: user local networks, cloud infrastructure, and enterprise internal networks. The system employs SSH reverse tunnels, cryptographic authentication and encryption, ensuring that neither attacker nor platform operator cannot access or decrypt customer data. Key security features including browser-based encryption, and multi-layered verification are examined, demonstrating how the architecture maintains complete data privacy and enterprise control.

Schema

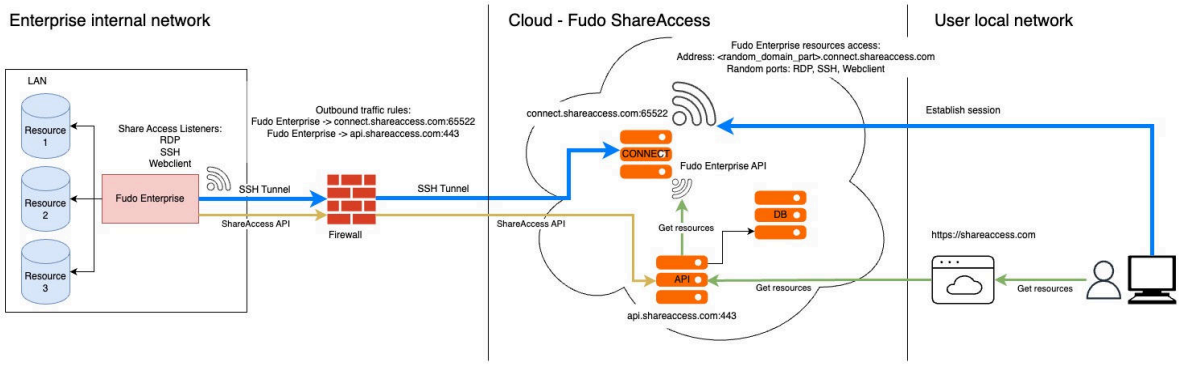

Figure 1: FSA_FE_diagram_v2.jpg

User Local Network

- **User**: A person who accesses Fudo ShareAccess through a web browser from their local network. After logging in, the user can view a list of available resources and use the provided data to establish a connection through the Connect component in the cloud infrastructure.
- **Browser**: A web browser running on the user's local machine, used to access the Fudo ShareAccess panel. All authentication and authorization processes take place via the browser, which communicates with the API component in the cloud.
  - Encryption and decryption of data occur locally in the browser, ensuring a **zero-knowledge architecture** when accessing organization-specific resources.

Fudo Security Sp. z o. o.
Al. Jerozolimskie 178
02—486 Warszawa, Poland
NIP 52425217
sales@fudosecurity.com2

Cloud Infrastructure

• **Fudo ShareAccess**

The core infrastructure of the Fudo ShareAccess product, which includes the following components:

- **API**: The main interface for communication between the frontend, database, and Fudo Enterprise instances.
- **Connect**: A component responsible for managing tunnels between Fudo Enterprise and Fudo ShareAccess, allowing secure communication via randomly generated ports.
- **Database (DB)**: Stores information about Fudo ShareAccess (FSA) accounts, organizations, and user invitations. Data is logically separated for security.

Component Details

• **Connect**

Establishes and maintains secure tunnels between Fudo Enterprise and Fudo ShareAccess. It generates random ports for communication and connection purposes. Through this tunnel, Fudo ShareAccess can communicate with Fudo Enterprise and forward user requests from the API to the relevant Fudo Enterprise API.

• **Database (DB)**

Stores data related to user accounts, organizations, and invitations. Each organization's data is logically separated to ensure security.

• **API**

Provides an interface for the frontend to interact with the database and Fudo Enterprises.

- Allows users to log in and retrieve a list of available resources.
- Generates secure connection secrets that can be used via the Connect component.

Fudo Security Sp. z o. o.
Al. Jerozolimskie 178
02—486 Warszawa, Poland
NIP 52425217
sales@fudosecurity.com3

Enterprise Internal Network

- **Firewall**: A set of rules that must be configured to allow Fudo Enterprise to connect to Fudo ShareAccess.
  - The Connect and API components operate on separate addresses and ports.
  - When a Fudo Enterprise instance joins Fudo ShareAccess, it retrieves necessary connection details via the API and establishes tunnels using the Connect component.
  - **No inbound connections** from Fudo ShareAccess to the enterprise network **are required—all communication is initiated by Fudo Enterprise**.
- **SSH Tunnel**: A secure communication channel used for session establishment and API communication.
  - When Fudo Enterprise is added to Fudo ShareAccess, it creates a **reverse tunnel** to the Connect component.
  - For each Fudo Enterprise, random ports are used for **SSH, RDP, WebClient, and API communication**. Those ports do not change for the whole time when Fudo Enterprise is paired with FSA.
  - On Connect there is a firewall running that keeps tunneled ports closed to the public. The port is opened for the user after he requests a connection in the FSA interface and successfully generates the OTP. The port is open only for a few minutes and only for the source IP of the computer on which user is logged in.
    - Note that if HTTPS connections (to port 443) are proxied through a different machine and other ports are not proxied, the user won't be able to establish connection through tunneled port.
- **Fudo Enterprise**: A standalone instance of Fudo Enterprise that integrates with Fudo ShareAccess.
  - Manages **resource access** configurations.
  - Stores **session recordings** of user interactions with resources.

Zero-knowledge

Fudo ShareAccess is built on a **zero-knowledge architecture**, ensuring that **user authentication, resource access, and session establishment** remain private and secure. Neither Fudo Security nor any third party can access or interpret sensitive data.

The entity that is considered authorized to session recordings is the organization that owns Fudo Enterprise which serves resources.

Note: Do not mistake zero-knowledge as understood in this document with zero-knowledge proof known from mathematics.

How It Works

Secure Communication with SSH Tunnels

- Fudo Enterprise establishes a **secure SSH tunnel** to the Connect component in the cloud.
- All **API requests, session data, and traffic** pass through this tunnel, ensuring **end-to-end encryption** between the user and their resources.
- Since the connection is initiated from Fudo Enterprise, no inbound access to the internal infrastructure is required.

User Authentication & Data Protection

- **Authentication is performed locally in the user's browser.** The user's credentials are never transmitted to Fudo Security in plaintext. Instead, the system derives a secure key using PBKDF2(master password, nonce) with 600,000 iterations, which is then sent along with the user's email for authentication.
- Upon logging in, the user retrieves their **private key** encrypted with AES-256-GCM using a key derived from master password. The private key is used for **digitally signing requests**.
- **All API requests are signed** before being sent to Fudo ShareAccess to prevent tampering. The requests are valid for a small time window to prevent replay attack.

Verifying Access & Establishing Sessions

- The Fudo ShareAccess API and Fudo Enterprise API work together to validate user requests.
- Only **authorized resources** are returned, based on signature verification.
- Secure **session secrets** are generated dynamically and also encrypted, ensuring that only authenticated users can establish connections.

Ensuring Absolute Data Privacy

- **Fudo Security never has access** to the OTP used to login to the target Fudo Enterprise. The OTP is encrypted on Fudo Enterprise and decrypted in user browser.
- Data passing through **SSH tunnels** remains **fully encrypted** and **unreadable**.
- Within tunnels, FSA allows establishing sessions with resources using only secure protocols: SSH, RDP over TLS and HTTPS for WebClient. SSH/TLS termination occurs on Fudo Enterprise.
- **No resource information or session data** is stored in the Fudo ShareAccess database – everything is retrieved on-demand.

Why Zero-Knowledge Matters?

With this architecture, **Fudo Security acts only as a communication bridge** – it cannot see, access, or store sensitive organization data. This guarantees **complete privacy and security** for organizations using Fudo ShareAccess.

Fudo Security Sp. z o. o.
Al. Jerozolimskie 178
02—486 Warszawa, Poland
NIP 52425217
sales@fudosecurity.com6

Authentication

Fudo ShareAccess implements an **authentication module** based on **cryptographic keys** and **request signing**. When a user creates an account, **cryptographic keys for signing and encryption** are generated **directly in the local browser**. Keys are stored in database AES-256-GCM encrypted, which makes them inaccessible for anyone other than user.

Authentication Process

- **Decryption takes place in the local browser** – never on the server.
- The user's password, combined with **hashing and salting mechanisms**, allows access to their **private key**.
- Every request sent to Fudo ShareAccess is **digitally signed** using this private key.
- The system **verifies these signatures** to determine whether the user is authorized to access data.
- Requests for resources and Fudo Enterprise interactions undergo an additional verification step on Fudo Enterprise itself.

Key Fingerprint Verification

- Each user has a **unique fingerprint** for their cryptographic keys, visible to both **themselves and organization administrators** which manage resources access.
- When inviting a user, an **admin can verify the fingerprint** to ensure they are communicating with the correct individual.
- If a user provides a fingerprint, the admin can **cross-check it** with the fingerprint stored in Fudo Enterprise, confirming that the user is properly authenticated.

Technical Details

Fudo ShareAccess utilizes multiple cryptographic algorithms for **authentication, encryption, and key management**:

- **ECDSA keys** – Used for **request signing** to authenticate user actions.
- **RSA-OAEP keys** – Used for **encrypting OTP**, used to authenticate against Fudo Enterprise instance, within the protocol that will be recorded (SSH/RDP) ensuring secure data transmission.
- **AES-256-GCM** – Used for **symmetric encryption** of stored keys, providing strong security.
- **PBKDF2** – Used for **deriving symmetric keys from user passwords**, strengthening key protection with hashing and salting.

By combining **local decryption, cryptographic signing, and layered encryption techniques**, Fudo ShareAccess **ensures strong security, privacy, and zero-knowledge authentication** while keeping **user credentials fully protected**

Fudo Security Sp. z o. o.
Al. Jerozolimskie 178
02—486 Warszawa, Poland
NIP 52425217
sales@fudosecurity.com7