emocha®

REVOLUTIONIZING PUBLIC HEALTH

Security & HIPAA Compliance White Paper

CONFIDENTIAL [Not intended for distribution]

emocha Security & HIPAA Compliance

Security Overview

The emocha platform is a suite of patient engagement and medication adherence applications, with both mobile (iOS / Android) and web components. All emocha applications comply with HIPAA regulations on how to handle protected health information (PHI), including but not limited to secure encryption of data, access controls, and industry-standard best practices. A robust role-based permission system limits system access to only authorized, authenticated users to ensure the need-to-know basis of PHI. All PHI is encrypted both in-flight and at-rest, and all access to, or modification of, patient data and system configuration is logged complying to both HIPAA and IRB requirements. The server infrastructure is secured from both physical and remote access.

Access

Access to the system is managed via password-protected user accounts. User passwords are never stored in clear text, only as a one-way encrypted digest, and are never visible to any user including emocha system administrators. The system includes rules to require that users create a complex password (with configurable minimum length, and requirements for special characters, numbers, etc.), and requires that user passwords be changed periodically. All access attempts, successful or otherwise, are logged. Repeated failed attempts result in the account being locked, and may only be unlocked by an administrator.

User Roles and Administration

User accounts are defined by a set of role-based permissions and only users with elevated permissions are capable of modifying a user's access. The system's position towards any user action is "default-deny". That is, unless a user has specifically been granted the right to perform an action, via a permission they've been explicitly granted by an administrator, the action is not permitted. An interface is provided both to add and remove roles from user accounts, and also to define new roles, or add and remove permissions from existing roles. Multiple roles can be assigned to the same user.

Input Validation

All input to the system is checked for validity before being processed. Validation is both done on the client and server sides; client-side as a user-experience convenience and server-side for data validation. The backend system assumes all incoming data to be tainted and will not use or store any data until validation is complete. This validation (based on formatting, length, range, etc.) is supported by the check for malicious intent (XSS attacks, SQL injection, etc.). Most parameters are marked as required and an absent or malformed required parameter or a present but malformed optional parameter will result in the entire request being declared invalid returning an error.

Authentication & Authorization

Authentication is managed with a username and password (adhering to the emocha password standards). Users are authorized to perform only the actions explicitly granted to them by the roles they have been assigned. Before allowing any user interaction with the system (for instance viewing patient information, creating a new user account, scheduling an appointment), the user's permissions are checked to determine if they are allowed to perform the specified action and whether they're allowed to access the object in question. A given user may be configured to only be allowed access to view or modify certain patients.

In the event that a user has forgotten their password, they can request a password reset for their username. This will generate a message to the email address associated with that account containing a time-limited single-user token which can be used to enter a new password.

Session Management & Timeouts

Every interaction with the system, with a few exceptions such as logging in and recovering a lost password, requires a valid, recent, session token which is returned as part of a successful authentication.

Session identifiers are stored as encrypted cookies on the device or browser, and chosen from a large, random, address space. These are not predictable and any modification of the local value will invalidate the session. No user-provided data (other cookies, roles/permissions, etc.) is used by the system and is ignored if provided (other than the session identifier). The backend fetches that

emocha Security & HIPAA Compliance

information from the database which is the trusted store of information.

Sessions time out after a configurable period of inactivity, for which the default is 5 minutes. When timeout happens, the user must re-authenticate to continue interacting with the system.

Encryption

The emocha platform uses two main kinds of encryption: in-flight and at-rest.

In-flight encryption refers to the encryption of all data while being transmitted. Data being sent from a client, whether web-based or a mobile application, is sent over a secure HTTPS connection secured by a 2048-bit SSL certificate. We audit our SSL configuration regularly, ensuring that system configuration is as up-to-date as possible. All connections between the database and application servers are made over SSL/TLS, using the same 2048-bit certificate.

At-rest encryption means that all protected health information (PHI) in the database and disk is always stored encrypted. This includes any record of a user, anything in the error log or audit log tables, any patient data, and all information submitted including video files or GPS coordinates. The encryption scheme uses the Advanced Encryption Standard (AES) algorithm of at least 256 bits, with the ability to revoke and issue new keys as needed. Data being sent from mobile devices is encrypted on the device as soon as it has been collected. Data is then transmitted to the server over a secure HTTPS channel and deleted from the device as soon as receipt of the transmission is confirmed. Photos or videos being recorded are stored on the application's partition of the SD card or internal storage, and not visible in the device's general media gallery applications.

When retrieving any data from the database, the encrypted data is fetched by the application, then decrypted before being sent to the client.

Encryption / decryption keys are housed on a separate server and only accessible through a highly-restrictive API, which is not directly reachable from the database server. Keys are only stored in memory on the application server and never in permanent files written to disk. Effectively, the database cannot decrypt its own data; even in the event of the server being compromised and a malicious party acquiring an export of the data, PHI will remain secure.

Each customer application's data (patients, checkins, laboratory test result data) resides in a separate database or schema and is completely invisible to other customers.

Audit Logs

Any viewing or modification of the system, or patient data, is logged in a persistent and unmodifiable database. Audit trail records include but are not limited to the action being taken, the user who initiated the action, the date and time, and, in the case of modifications, both the old and new values. These logs are available to be searched with numerous sorting and filtering options on the administrative interface. In addition, nothing is ever deleted in the system; data is "soft-deleted" via marking with a flag that will hide the record during normal operations, but leave it easily recoverable if needed.

Infrastructure, Hosting Environment, and Backups

emocha's servers are hosted "in the cloud" at secure data centers. Physical access is extremely limited if not impossible. All servers are single-tenant, and there is no shared hardware between emocha and any other entity.

Network access to any server is limited to the specific port and IP ranges needed for the platform to function (for instance, the front-facing load balancers allow access via HTTPS from anywhere but the database servers only allow SSL/TLS access over port 3306 from the application servers they specifically support), and SSH access is only permitted from the emocha office itself. Login credentials are managed by a combination of a strong password and private keys. Keys are only distributed via direct USB storage and never sent over any network. Login passwords are required to adhere to the emocha Password Policy, explained above. Access to production servers is only given to employees with a demonstrable need for access, who are needed to provide production support for the service.

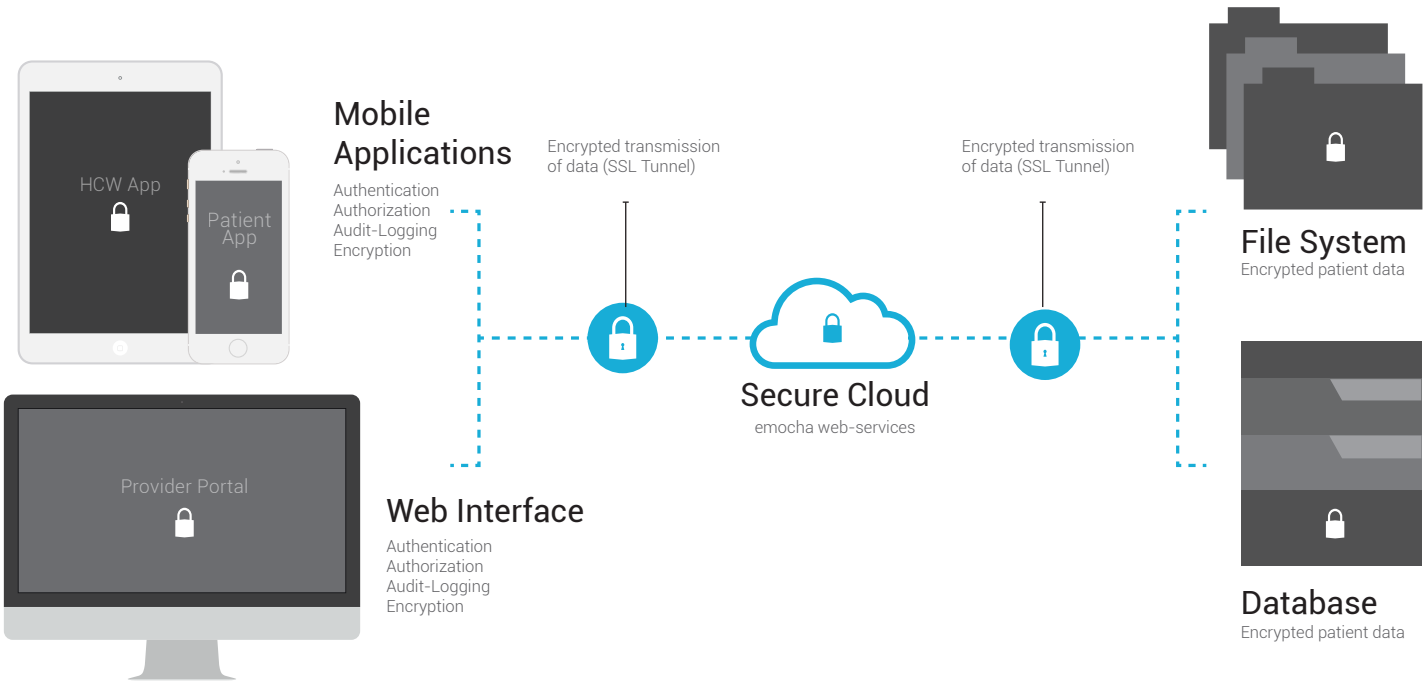
Remote console access, where these settings are managed, requires a two-factor authentication token in addition to a strong password.

emocha Security & HIPAA Compliance

Accounts used to connect to servers are per-user rather than a shared “root” account. This allows for individual users to be managed, or access revoked, without compromising other accounts.

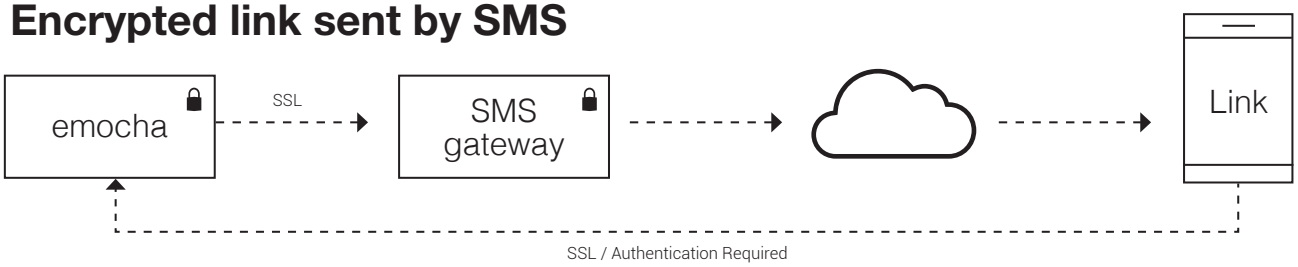
Backups are taken regularly; nightly backups are made of the entire encrypted database and snapshots of the entire server disk are taken, including any encrypted file uploads. These are kept available across a rolling 30-day window in case they need to be rolled back to. All data is replicated across multiple servers, in near real-time, to ensure availability. We also run monitoring software that checks, at all times, whether the service is available and functioning, and have a system in place to page on-call support personnel if needed.

Security Architecture Diagram

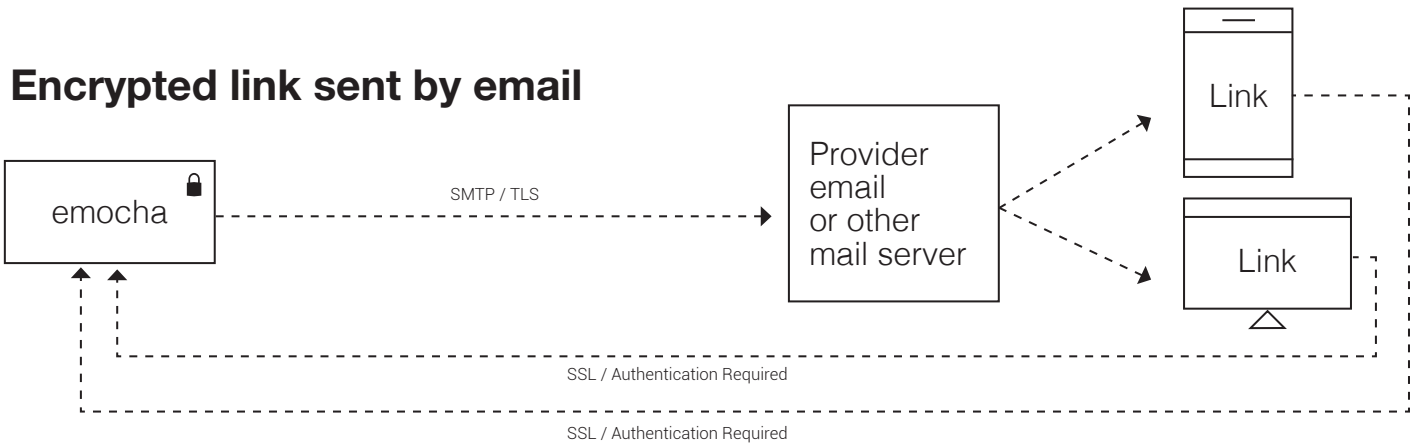


emocha Security & HIPAA Compliance

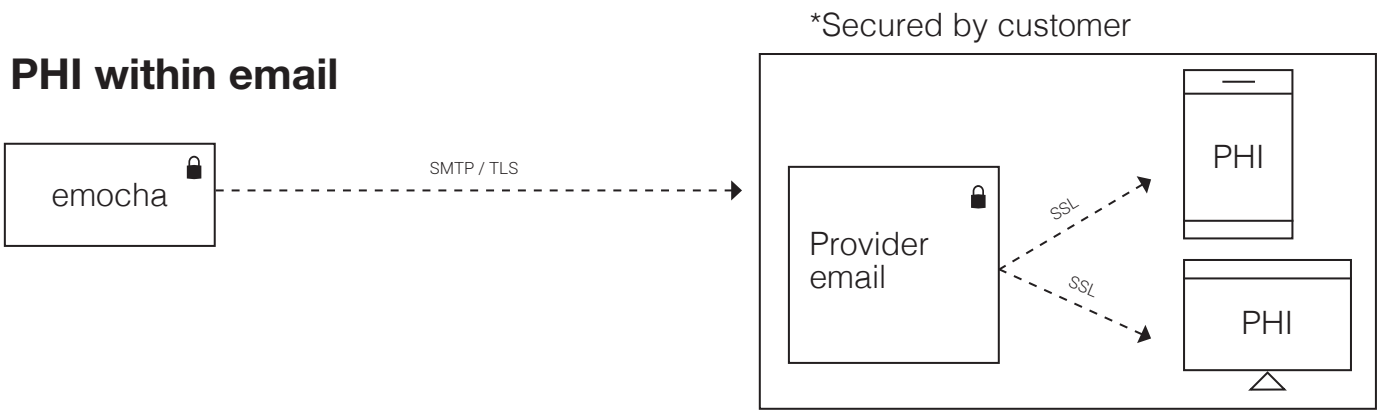
Encrypted link sent by SMS



Encrypted link sent by email



PHI within email



In-App Notifications