IronCircles, Inc

Privacy and Security

June, 2023

Introduction

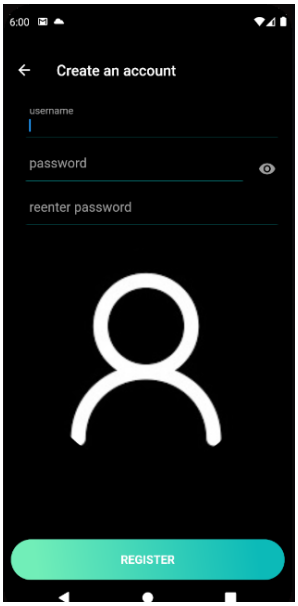
Privacy and security are at the heart of the IronCircles platform. Each feature and function was designed to protect the user's privacy and ensure the safest platform possible based on modern standards. Encryption protocols are constantly evolving to stay ahead of emerging threats. As security professionals, we know there is no such thing as the perfect system. Instead, IronCircles was built to leverage the latest information security best practices and offer a platform that we ourselves are confident in to store sensitive data.

Table of Contents

I. Privacy

i. Anonymous

IronCircles does not collect personal information from users of the platform. Registering users are only required to create a unique username and enter a passphrase. Selecting an avatar is optional.



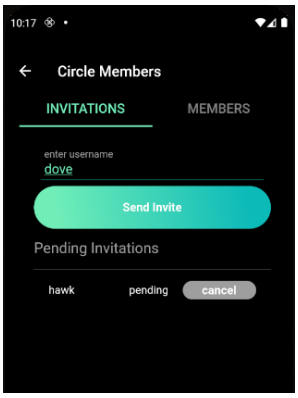
No name, email, phone number or any other type of identifiable information is collected.

The platform will ask the user to confirm they are over 16. If they answer no they are not able to register.

IronCircles does not publish a list of usernames inside or outside of the platform.

ii. Invitation Only

Users can only be added to Circles through invitations. A user must know the username of another user in order to invite them to a circle.



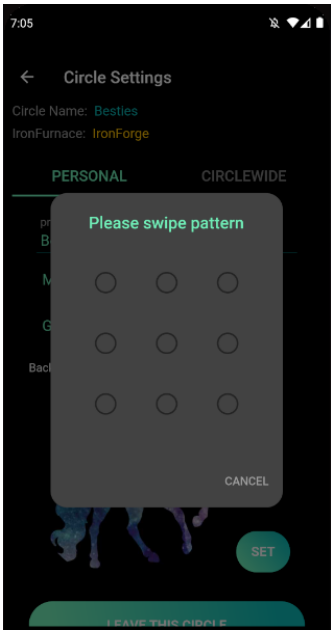
Circles require a vote before a new invitation is sent. The vote is either Unanimous (default) or Majority Rules based on that Circle's settings.

Members can also be removed from Circles by the same voting mechanism.

The platform provides a blocked list to stop unwanted circle invitations.

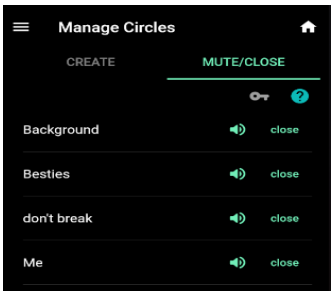
iii. In App Privacy

IronCircles offers additional optional layers of privacy protection within the app.



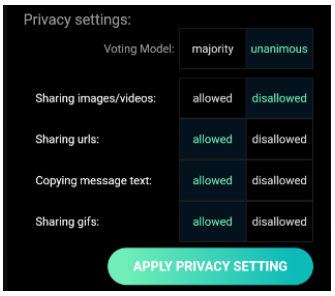
Guarded Circles are Circles that are visible on the home screen but must be unlocked with a swipe pattern before entering. Message contents from Guarded Circles are not visible in the Library.

Circles can be permanently unguarded by turning off the setting after access to the Circle has been granted.



Circles can also be Closed or Muted. Muting a Circle prevents push notifications and Closing a Circle prevents push notifications as well as removes the Circle from the home screen until the Circle is Opened.

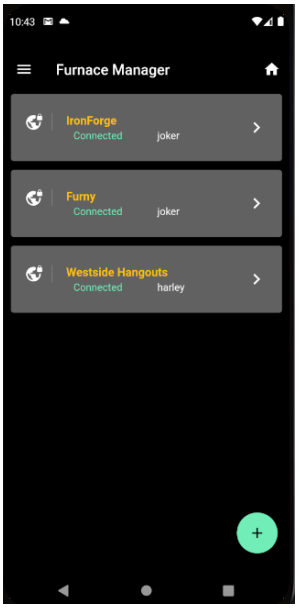
iv. Privacy Settings



The following privacy settings are adjustable by the Circle. A request to change these settings will kickoff a vote for the Circle. The type of vote can also be changed from unanimous to majority rules (also through a vote).

- Sharing images/videos outside the circle (off by default)
- Sharing urls outside the circle
- Copying message text
- Sharing gifs

v. Decentralization



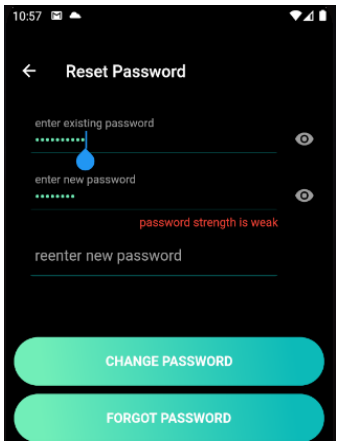
A major feature of this platform is the ability to decentralize. The hosted cloud version of IronCircles is called the IronForge. All users have the option of connecting to the IronForge (or not).

The IronForge platform has been containerized and is available for self hosting by users who choose to do so. Self hosted instances are called IronFurnaces. Users can be connected to the IronForge, host their own circles on an IronFurnace, and connect to an unlimited number of friend's IronFurnaces seamlessly at the same time.

IronFurnaces can be added, deleted, connected, or disconnected. Users may opt to leverage different usernames and password combos on each furnace they connect to for obfuscation of identity.

II. Security

i. Password Management



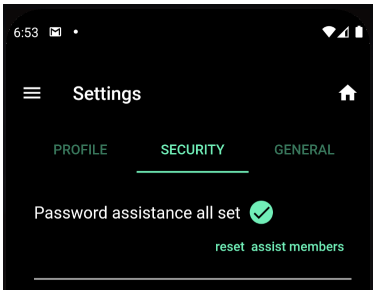
IronCircles uses traditional passwords to authenticate users and allow them to connect on multiple devices.

The password/pin combo is salted and hashed before being stored.

The password and pin combo is ran through a KDF (Hkdf: Hmac.sha256) to generate a backup key which is used to encrypt chat history, if the user elects to.

Once authenticated, a security token is created for the user. Upon expiration of the token, the user must reauthenticate.

Password resets take into account that users are anonymous.



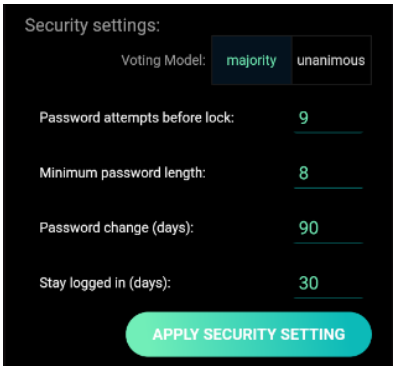
Users are prompted to select between 1 and 4 other members who can assist with a password reset.

If the user kicks off a reset, each password assist member is sent an equal length fragment of a reset code. The user can assemble the reset fragments together, enter the reset code, and create a new password. While selecting only 1 other member is allowed, the user is warned that the 1 member will get the entirety of the reset code.

Users are also prompted with password complexity suggestions when they register and/or reset their passwords. Passwords are validated against dictionaries and commonly used lists.

ii. Security Settings

The following security settings are adjustable by the circle. A request to change these settings will kickoff a vote for the Circle. The type of vote can also be changed from unanimous to majority rules (also through a vote).



- Password attempts before lockout
- Minimum password length
- Days before password change required
- Stay logged in (days)

A user's security settings are set according to the strictest Circle they belong to. For example, if a user has a 8 character password and they are invited to a Circle that requires 12 characters, they are prompted to reset their password before they can join.

iii. E2E Encryption (Forward Secrecy)

IronCircles implements a Forward Secrecy algorithm to protect user data. Each message sent from the platform is encrypted with a new secret key, nonce, and MAC signature In the event that a single message is compromised, all past and future messages are still secure.

IronCircles uses a Double Ratchet approach to achieve Forward Secrecy.

Keychain Generation

A user can join a circle by creating a new one or accepting an invitation to an existing one. Upon joining a circle, two new keychains are created on behalf of the user. The first keychain is for receiving messages to that Circle. A ECDH keypair is generated for the receiving keychain and the public key is made available to other members of the Circle. No one outside of the Circle has access to the public key. The second keychain is for sending messages. A ECDH keypair is not generated for that chain until the user sends a message.

Sending a Message

Steps that occur when a message is sent:

1. First ratchet - A secret is generated (the Message Key) and the message contents are encrypted with the secret using **XChaCha20 256-bit** encryption, a unique nonce, and then signed using **HMAC-SHA256** to validate the contents have not been tampered with..
2. Second ratchet - for each receiver in the Circle:
    3. Receiver's advertised receiving public key (for that Circle) is pulled from the API.
    4. Sender generates a new Sending ECDH keypair.
    5. A shared secret is calculated from the Sender's new private key and the Receiver's advertised receiving public key (for that Circle).
    6. The shared secret is used to encrypt the Message Key.
    7. The public key from the Sending ECDH keypair, the Receiving KeyIndex of the advertised Receiving public key, encrypted Message Key, and encrypted Message Contents are sent to members of the circle.

Receiving a Message

Steps that occur when a message is received:

1. Platform matches the message's included Receiving KeyIndex with a ECDH keypair on the device.
2. The private key from that keypair is used with the message's Sender public key to calculate the shared secret.
3. The shared secret is used to decrypt the Message Key.
4. The Message Key is used with the nonce and MAC to decrypt the Message.
5. A new Receiving ECDH keypair is generated, the Receiving keychain is ratcheted, and the new public Receiving key is advertised to the Circle.

A Circle member's current Receiving public key is not available after 90 of no activity. A new keypair is issued upon next login. Messages will not be sent to the user in the meantime.

Protection Against MiTM Attacks (version 1.1.16+)

Sender Key Validation - At the initiation of a conversation (in a Circle or DM), the connected users exchange public Identity keys. When a user sends a message, the message is signed using their private Identity key for the device they are on. The receiver validates the signature with the public Identity key that was sent during the beginning of their conversation history.

If a message cannot be verified with the public key, the receiver is warned that the sender cannot be verified and may be on a new device.

Receiver's are also warned when an authenticated user they are connected to registers a new device with the API (a message is broadcast to all Circles/DMs). In addition to the broadcast message, the first message sent on the new device will fail verification and the receiver will be warned the sender may be on a new device.

View Sender Identity Public Keys - The receiver can visually validate the sender's keys by clicking on their profile and matching the message Identify key to a corresponding public key/device combo.

Non-repudiation - If the signature verification passes for a received message, the receiver knows the sender has been verified and validated.

User Keychain

When a user registers on the platform, the user is assigned a User Keychain and associated long lived keys. Keys added to this chain are only used to encrypt user preferences and settings, never message contents.

Encryption in Transit

In addition to end-to-end encryption, all data transmitted from the platform is encrypted in transit using HTTPs.

IronForge - Encryption at Rest

The IronForge takes a zero trust approach and encrypts the data at rest even though it has already been E2E encrypted using Forward Secrecy.

iv. Cryptography Key Management

IronCircles stores Identity, User, Receiving, and Sending keychains in protected storage on the local device until the keys are no longer needed.

Uninstall

Uninstalling the IronCircles app removes all keychains. If a user uninstalls and reinstalls, new keychains and Identify keys are generated. They will not have message history unless they have Chat History Backup turned on.