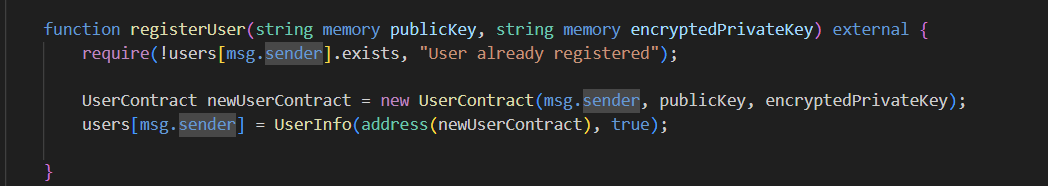
**1. Factory Contract**

Factory contract là contract chính quản lý người dùng và nhóm.

**1.1 Đăng ký người dùng (registerUser)** 

* Người dùng mới gọi registerUser(publicKey, encryptedPrivateKey)
* Hợp đồng kiểm tra xem người dùng đã tồn tại chưa (users[msg.sender].exists)
* Nếu chưa, hệ thống tạo một UserContract mới
* Lưu địa chỉ UserContract vào users mapping
* FE tạo cặp key r mã hóa privatekey r gọi hàm

**1.2 Tạo nhóm (createGroup)**

A black screen with colorful text

AI-generated content may be incorrect.

* Người dùng gọi createGroup(groupPublicKey, encryptedGroupPrivateKey)
* Hợp đồng kiểm tra xem người dùng đã đăng ký chưa (users[msg.sender].exists)
* Tăng groupCounter
* Tạo một GroupChat mới với admin là người gọi
* Lưu thông tin nhóm vào groups mapping
* FE tạo cặp key r mã hóa privatekey bằng publickey của chính nó r gọi hàm

**1.3 Lấy địa chỉ hợp đồng**A black background with white text

AI-generated content may be incorrect.A black screen with text

AI-generated content may be incorrect.

* getGroupContract(groupId): Trả về địa chỉ hợp đồng của nhóm nếu tồn tại
* getUserContract(user): Trả về địa chỉ hợp đồng cá nhân của user nếu tồn tại

**2. UserContract**

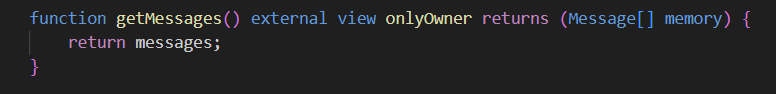
Mỗi người dùng có một hợp đồng cá nhân để quản lý tin nhắn.

**2.1 Gửi tin nhắn (sendMessage)** A black background with white text

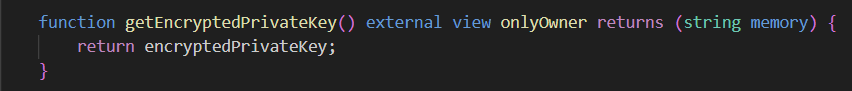
AI-generated content may be incorrect.

* Người dùng gọi sendMessage(encryptedMessage)
* Tin nhắn được lưu vào messages array
* Emit sự kiện MessageReceived
* FE mã hóa tin nhắn bằng publickey người nhận r gọi hàm



**2.2 Lấy tin nhắn (getMessages)** 

* Chỉ chủ sở hữu có thể gọi getMessages()
* Trả về danh sách tin nhắn của người dùng
* Lấy privatekey ở 2.3 giải mã bằng publickey của người dùng => giải mã tin nhắn bằng privatekey đã giải mã

**2.3 Lấy khóa riêng (getEncryptedPrivateKey)** 

* Chỉ chủ sở hữu có thể lấy encryptedPrivateKey

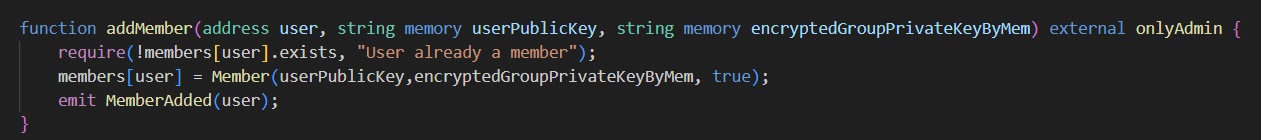
**2.4 Đánh dấu tin nhắn đã đọc (markMessageAsRead)** A computer code on a black background

AI-generated content may be incorrect.

* Chỉ chủ sở hữu có thể gọi markMessageAsRead(index)
* Kiểm tra chỉ mục hợp lệ và tin nhắn chưa được đọc
* Đánh dấu tin nhắn là "đã đọc"
* Người dùng đọc tin nhắn nào r thì gửi hàm đã đọc tin nhắn đó

**3. GroupChat**

Mỗi nhóm có một hợp đồng riêng để quản lý thành viên và tin nhắn.

**3.1 Thêm thành viên (addMember)** 

* Chỉ admin có thể gọi addMember(user, userPublicKey, encryptedGroupPrivateKeyByMem)
* Kiểm tra nếu user chưa phải thành viên
* Thêm user vào members mapping
* Emit sự kiện MemberAdded
* Lấy publickey của đỉa chỉ người dùng , lấy khóa riêng tư GR ở 3.4 , lấy khóa công khai của ADMIN=>giải mã khóa riêng tư của GR bằng khóa riêng tư của admin(giải mã privatekey của admin bằng publickey của admin trc) => mã hóa lại nó bằng khóa công khai người dùng r gọi hàmA screen shot of a computer screen

  AI-generated content may be incorrect.



**3.2 Gửi tin nhắn (sendMessage)** A screen shot of a computer

AI-generated content may be incorrect.

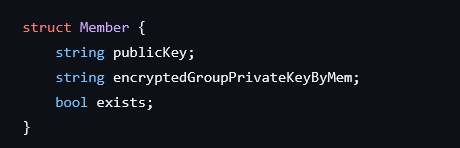
* Chỉ thành viên có thể gửi tin nhắn
* Tin nhắn được lưu vào messages array
* Emit sự kiện MessageSent
* Mã hóa tin nhắn bằng khóa công khai nhóm r gọi hàm

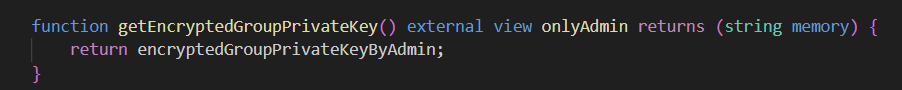


**3.3 Lấy tin nhắn (getMessages)** A black background with white text

AI-generated content may be incorrect.

* Chỉ thành viên có thể gọi getMessages()
* Trả về danh sách tin nhắn của nhóm
* Lấy privatekey mã hóa trong đã mã hóa của gr trong member giải mã bằng privatekey của của chính mình ở 2.3 (giải mã privatekey của chính mình bằng publickey trc)=> lấy privateGR đã giải mã để giải mã tin nhắn



**3.4 Lấy khóa riêng của nhóm (getEncryptedGroupPrivateKey)** 

* Chỉ admin có thể lấy encryptedGroupPrivateKeyByAdmin