Security programming

Overall review

The project has a clear structure, clear component responsibilities, and provides POC and configuration, which are reproducible and friendly. However, there are obvious backdoors in key strength verification and cross-server signature verification, which can be exploited once opened.

Strengths

1. Excellent engineering, clear module division, explicit logging, and a built-in PoC reduce testing costs.

2. Comprehensive protocol coverage, including user announcements/routes, session messages, and persistence paths.

3. Boot node mode, environment variable switches, and direct database verification.

Bugs and weaknesses

1. Setting client environment variables generates an RSA-1024 public key. The server does not perform minimum bit verification and registers the weak public key in the database. This makes identity forgery more difficult. Remediation suggestion: Keep the weak key switch on the client for offline testing only and disable it by default. At the same time, the server should absolutely distrust the client's bit count declaration and use the actual imported public key object for
   verification.

```
BACKDOOR_TRUST_GOSSIP=1 python3 server.py --introducer --host 127.0.0.1 --port 8
765
loaded existing keys
[keys] Loaded keys for server-0a213b0a →.keys/server-0a213b0a.priv.pem / .pub.p
em
[vault] SQLite database initialised and public channel ready.
[0a213b0a-2cfb-40d8-94c8-3363ca76d7ed] Listening on ws://127.0.0.1:8765


[0a213b0a-2cfb-40d8-94c8-3363ca76d7ed] New connection received.
[BACKDOOR] accepting unsigned USER_ADVERTISE for b6ac7cd4-01d9-4170-b5bd-6fffc9d
ccc16 (origin=bac7b965-16fe-40ab-80fc-384fe59f5712)
[gossip] Learned user b6ac7cd4-01d9-4170-b5bd-6fffc9dccc16 is on server bac7b965
-16fe-40ab-80fc-384fe59f5712
[gossip] Failed to decode pubkey for b6ac7cd4-01d9-4170-b5bd-6fffc9dccc16: Could
 not deserialize key data. The data may be in an incorrect format, it may be enc
rypted with an unsupported algorithm, or it may be an unsupported key type (e.g.
 EC curves with explicit parameters). Details: ASN.1 parsing error: short data (
needed at least 54 additional bytes)
```

2. When BACKDOOR_TRUST_GOSSIP=1 is set on the server, RSASSA-PSS signature verification is skipped, allowing unsigned or incorrectly signed user announcements to be accepted. This can compromise routing or identity, impacting message delivery. Recommended fix: Remove or disable this backdoor branch by default, and discard any messages that fail signature verification.

```
ygy@ygy-VirtualBox:~/Fangyu/Security-Programming$ python3 - << PY
import sqlite3, base64
from cryptography.hazmat.primitives.serialization import load_der_public_key

pad=lambda s: s + "="*(-len(s)%4)
uid, derb64 = sqlite3.connect("data_vault.sqlite").execute(
    "select user_id, pubkey from users order by rowid desc limit 1"
).fetchone()
bits = load_der_public_key(base64.urlsafe_b64decode(pad(derb64))).key_size
print("latest user:", uid, "key_bits:", bits)
PY
latest user: ba0978e1-2c76-4a1a-8bcd-e21368ee9863 key_bits: 1024
ygy@ygy-VirtualBox:~/Fangyu/Security-Programming$
```