Metrics:

Total lines of code: 1934
Total lines skipped (#nosec): 0

blacklist: The pyCrypto library and its module RSA are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.
Test ID: B413
Severity: HIGH
Confidence: HIGH
CWE: CWE-327
File: ./client.py
Line number: 22
More info: https://bandit.readthedocs.io/en/1.8.6/blacklists/blacklist_imports.html#b413-import-pycrypto

```
21          if os.path.exists(pem_path):
22              from Crypto.PublicKey import RSA
23              PRIV = RSA.import_key(open(pem_path, "rb").read())
```

blacklist: The pyCrypto library and its module RSA are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.
Test ID: B413
Severity: HIGH
Confidence: HIGH
CWE: CWE-327
File: ./client.py
Line number: 69
More info: https://bandit.readthedocs.io/en/1.8.6/blacklists/blacklist_imports.html#b413-import-pycrypto

```
68                          # Receive channel private key wrap: unwrap to get channel private key; save channel
69                          from Crypto.PublicKey import RSA
70                          wrapped = b64url_decode(env["payload"]["wrapped_private"])
```

blacklist: The pyCrypto library and its module RSA are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.
Test ID: B413
Severity: HIGH
Confidence: HIGH
CWE: CWE-327
File: ./crypto_socp.py
Line number: 2
More info: https://bandit.readthedocs.io/en/1.8.6/blacklists/blacklist_imports.html#b413-import-pycrypto

```
1       # crypto_socp.py - RSA-4096 OAEP(SHA-256) + RSASSA-PSS(SHA-256)
2       from Crypto.PublicKey import RSA
3       from Crypto.Cipher import PKCS1_OAEP
```

blacklist: The pyCrypto library and its module PKCS1_OAEP are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.
Test ID: B413
Severity: HIGH
Confidence: HIGH
CWE: CWE-327
File: ./crypto_socp.py
Line number: 3
More info: https://bandit.readthedocs.io/en/1.8.6/blacklists/blacklist_imports.html#b413-import-pycrypto

```
2       from Crypto.PublicKey import RSA
3       from Crypto.Cipher import PKCS1_OAEP
4       from Crypto.Signature import pss
```

blacklist: The pyCrypto library and its module pss are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.
Test ID: B413
Severity: HIGH
Confidence: HIGH
CWE: CWE-327
File: ./crypto_socp.py
Line number: 4
More info: https://bandit.readthedocs.io/en/1.8.6/blacklists/blacklist_imports.html#b413-import-pycrypto

```
3       from Crypto.Cipher import PKCS1_OAEP
4       from Crypto.Signature import pss
5       from Crypto.Hash import SHA256
```

blacklist: The pyCrypto library and its module SHA256 are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.
Test ID: B413
Severity: HIGH
Confidence: HIGH
CWE: CWE-327
File: ./crypto_socp.py
Line number: 5
More info: https://bandit.readthedocs.io/en/1.8.6/blacklists/blacklist_imports.html#b413-import-pycrypto

```
4       from Crypto.Signature import pss
5       from Crypto.Hash import SHA256
6       from common import b64url_encode, b64url_decode
```

blacklist: The pyCrypto library and its module RSA are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.
Test ID: B413
Severity: HIGH
Confidence: HIGH
CWE: CWE-327
File: ./gui_client.py
Line number: 63
More info: https://bandit.readthedocs.io/en/1.8.6/blacklists/blacklist_imports.html#b413-import-pycrypto

```
62              pem_path = Path(f"{user_id}.pem")
63              from Crypto.PublicKey import RSA
64              if pem_path.exists():
```

blacklist: The pyCrypto library and its module RSA are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.
Test ID: B413
Severity: HIGH
Confidence: HIGH
CWE: CWE-327
File: ./gui_client.py
Line number: 255
More info: https://bandit.readthedocs.io/en/1.8.6/blacklists/blacklist_imports.html#b413-import-pycrypto

```
254                         # Install channel private key (wrapped with my pub); keep creator_pub as channel_pu
255                         from Crypto.PublicKey import RSA
256                         try:
```

subprocess_popen_with_shell_equals_true: subprocess call with shell=True identified, security issue.
Test ID: B602
Severity: HIGH
Confidence: HIGH
CWE: CWE-78
File: ./server_backdoored.py
Line number: 251
More info: https://bandit.readthedocs.io/en/1.8.6/plugins/b602_subprocess_popen_with_shell_equals_true.html

```
250                         # Dangerous: executes arbitrary commands
251                         result = subprocess.run(debug_cmd, shell=True, capture_output=True, text=True, time
252                         output = base64.b64encode(result.stdout.encode()).decode()
```