

Summary: Some `libp2p` implementations (especially in **Rust**, **Python**, or **C/C++**) use or depend on **cryptographic libraries** like `gmp`, `libsodium`, or `curve25519-dalek`. On M2 (Apple Silicon), these dependencies can fail to compile due to:

- Architecture mismatches (`x86_64` vs `arm64`)
- Missing headers or libraries
- Incorrect environment configuration

You might see errors like:

```
fatal error: 'gmp.h' file not found
ld: library not found for -lgmp
error: gmp.h not found or gmp is not installed properly
```

Step-by-Step Resolution Guide

1. Install GMP Using Homebrew (Apple Silicon version)

```
brew install gmp
```

This installs GMP to:

```
/opt/homebrew/opt/gmp
```

! Important: If you're using an M1/M2/M3 chip, **do not** use Rosetta or Intel versions of Homebrew unless you have a good reason.

2. Set Environment Variables (Compiler & Linker Paths)

Add these to your terminal session:

```
export LDFLAGS="-L/opt/homebrew/opt/gmp/lib"
export CPPFLAGS="-I/opt/homebrew/opt/gmp/include"
export PKG_CONFIG_PATH="/opt/homebrew/opt/gmp/lib/pkgconfig"
```

Or add them to your shell profile:

```
# For zsh users
echo 'export LDFLAGS="-L/opt/homebrew/opt/gmp/lib"' >> ~/.zshrc
echo 'export CPPFLAGS="-I/opt/homebrew/opt/gmp/include"' >> ~/.zshrc
echo 'export PKG_CONFIG_PATH="/opt/homebrew/opt/gmp/lib/pkgconfig"' >> ~/.zshrc
source ~/.zshrc
```

3. For Rust (e.g., rust-libp2p)

If you're compiling a Rust project that uses `libp2p`, `curve25519-dalek`, or `rsa`, GMP may be needed by a transitive dependency (e.g., `num-bigint-dig`).

Try:

```
cargo clean
cargo build
```

If that fails:

```
PKG_CONFIG_PATH="/opt/homebrew/opt/gmp/lib/pkgconfig" \
cargo build
```

And ensure you're using a native `arm64` Rust toolchain:

```
rustup show active-toolchain
```

Use:

```
rustup update
```

4. For Python (e.g., gmpy2 or py-libp2p)

If the error is during `gmpy2` install (used in cryptographic number libraries):

```
pip install gmpy2 --global-option=build_ext \
  --global-option="-I/opt/homebrew/opt/gmp/include" \
  --global-option="-L/opt/homebrew/opt/gmp/lib"
```

Or use **conda** if you're working in a scientific/development context:

```
conda install -c conda-forge gmpy2
```

5. For Go (e.g., go-libp2p)

Most Go libp2p packages don't use GMP. But if you're using **cgo** or a crypto wrapper that needs GMP:

```
CGO_CFLAGS="-I/opt/homebrew/opt/gmp/include" \  
CGO_LDFLAGS="-L/opt/homebrew/opt/gmp/lib" \  
go build ./...
```

You can verify the library is found with:

```
pkg-config --libs gmp
```

Should return:

```
-L/opt/homebrew/Cellar/gmp/... -lgmp
```

6. Debugging and Verifying

 **Check architecture:**

```
uname -m
```

Should return:

```
arm64
```

 **Clean builds:**

Rust:

```
cargo clean && cargo build
```

Python:

```
pip uninstall gmpy2  
pip cache purge
```

Go:

```
go clean -cache
```



Example: Building Rust libp2p Project on M2

```
brew install gmp  
export LDFLAGS="-L/opt/homebrew/opt/gmp/lib"  
export CPPFLAGS="-I/opt/homebrew/opt/gmp/include"  
export PKG_CONFIG_PATH="/opt/homebrew/opt/gmp/lib/pkgconfig"  
cargo build
```

Still Broken?

If none of this works, please connect with us at libp2p discord:

1. The **exact error message**
2. The **command you're running**
3. Whether the project is **Rust**, **Python**, or **Go**
4. The GitHub repo URL, if available

----- Contribution Section (Arush from Huddle01)-----

GitHub Id: <https://github.com/arcinston>