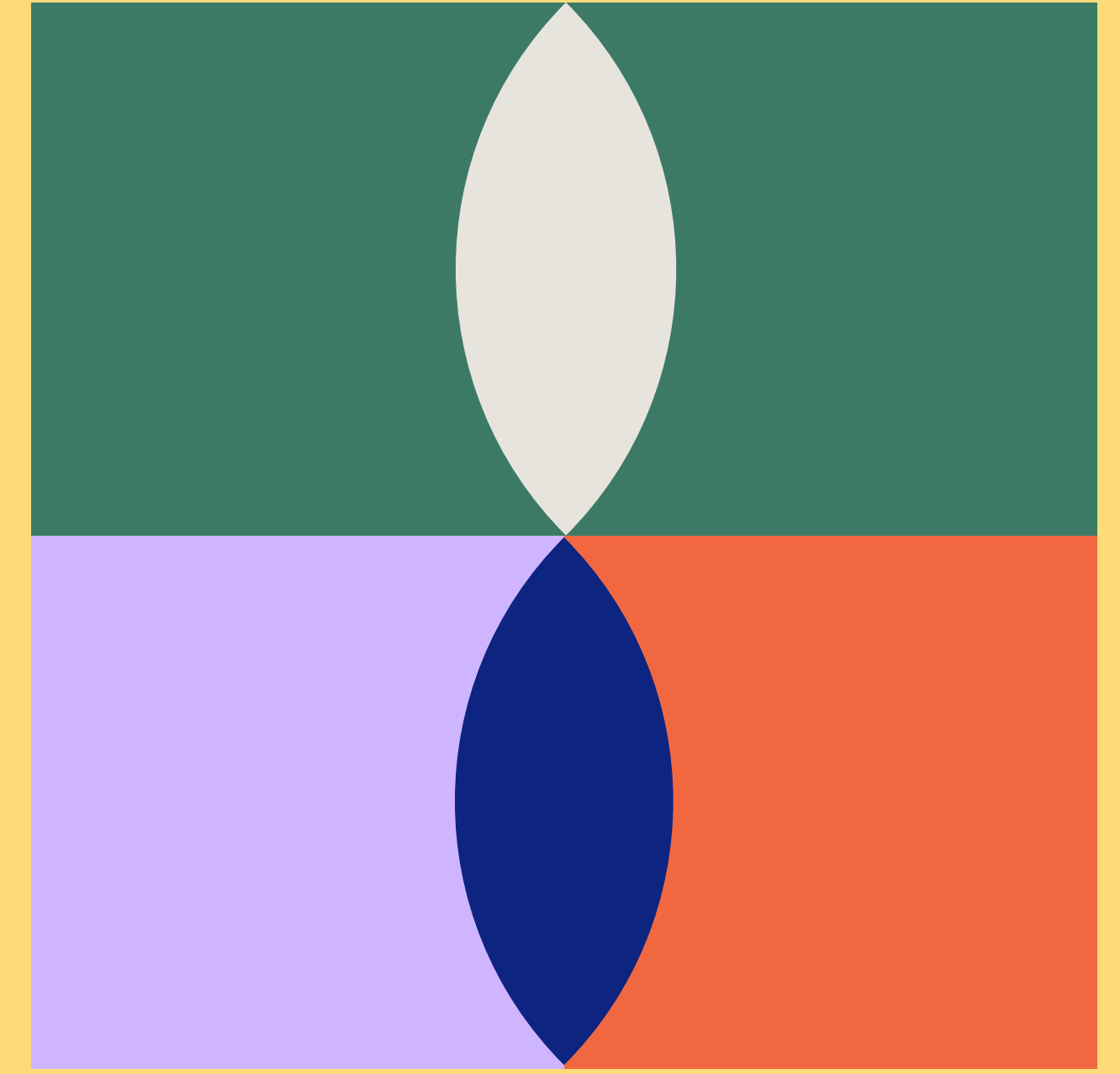


From Capturing Flags to Cryptography Consultancy

Giacomo Pope, NCC Group

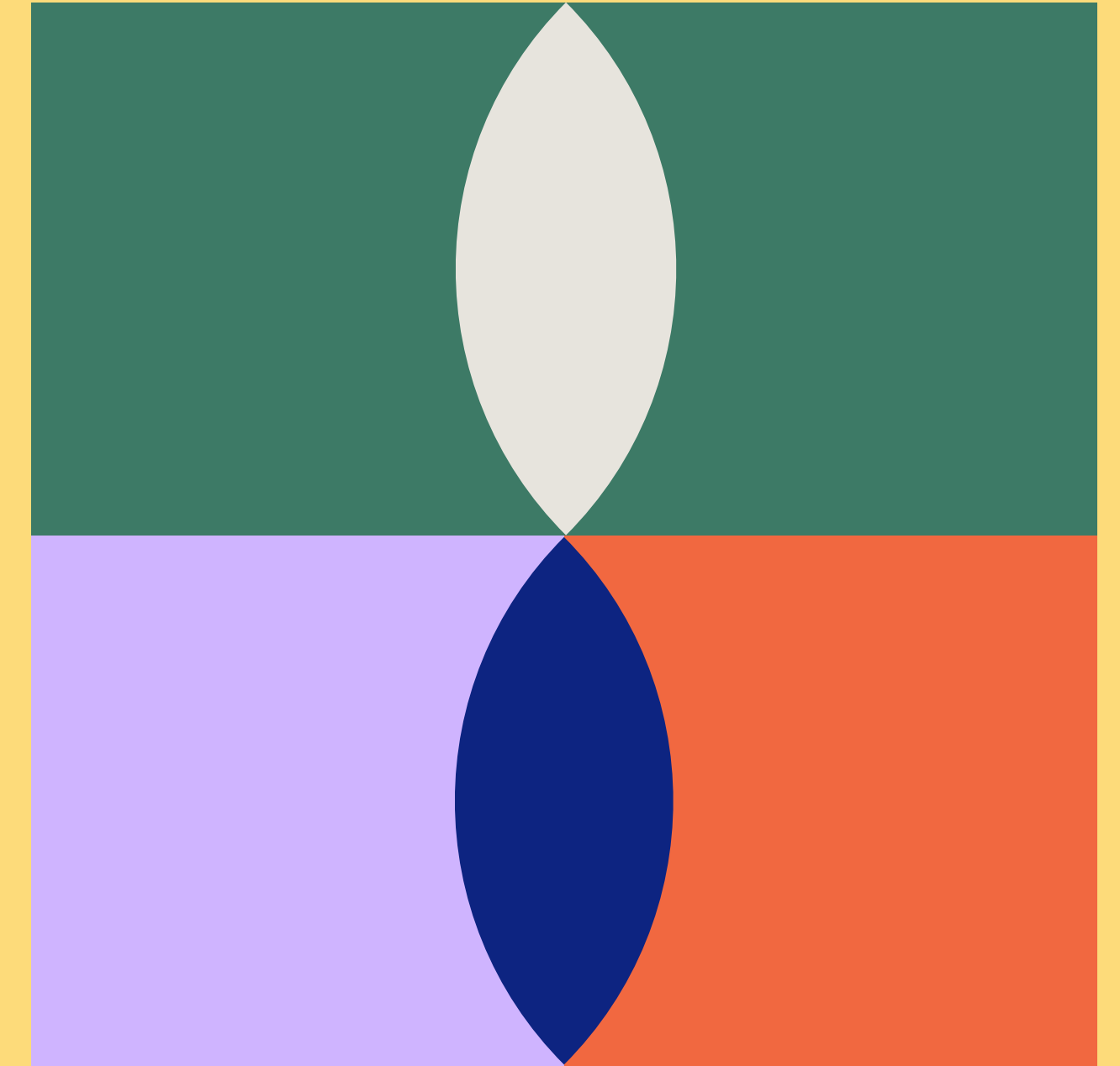
M0lecon, Politecnico di Torino, March 21st 2026

What's The Plan?



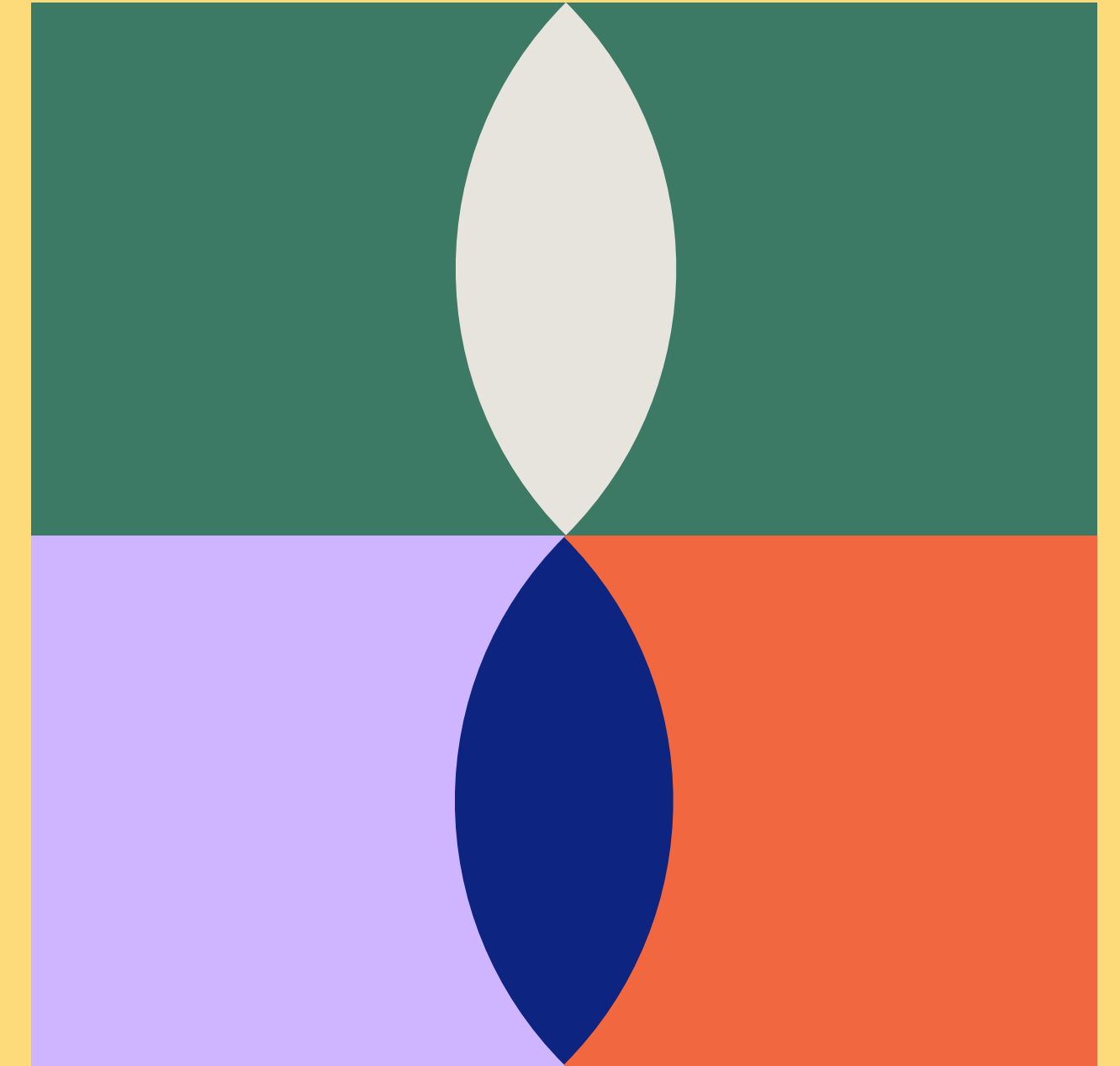
What's The Plan?

- We're here because we love CTFs



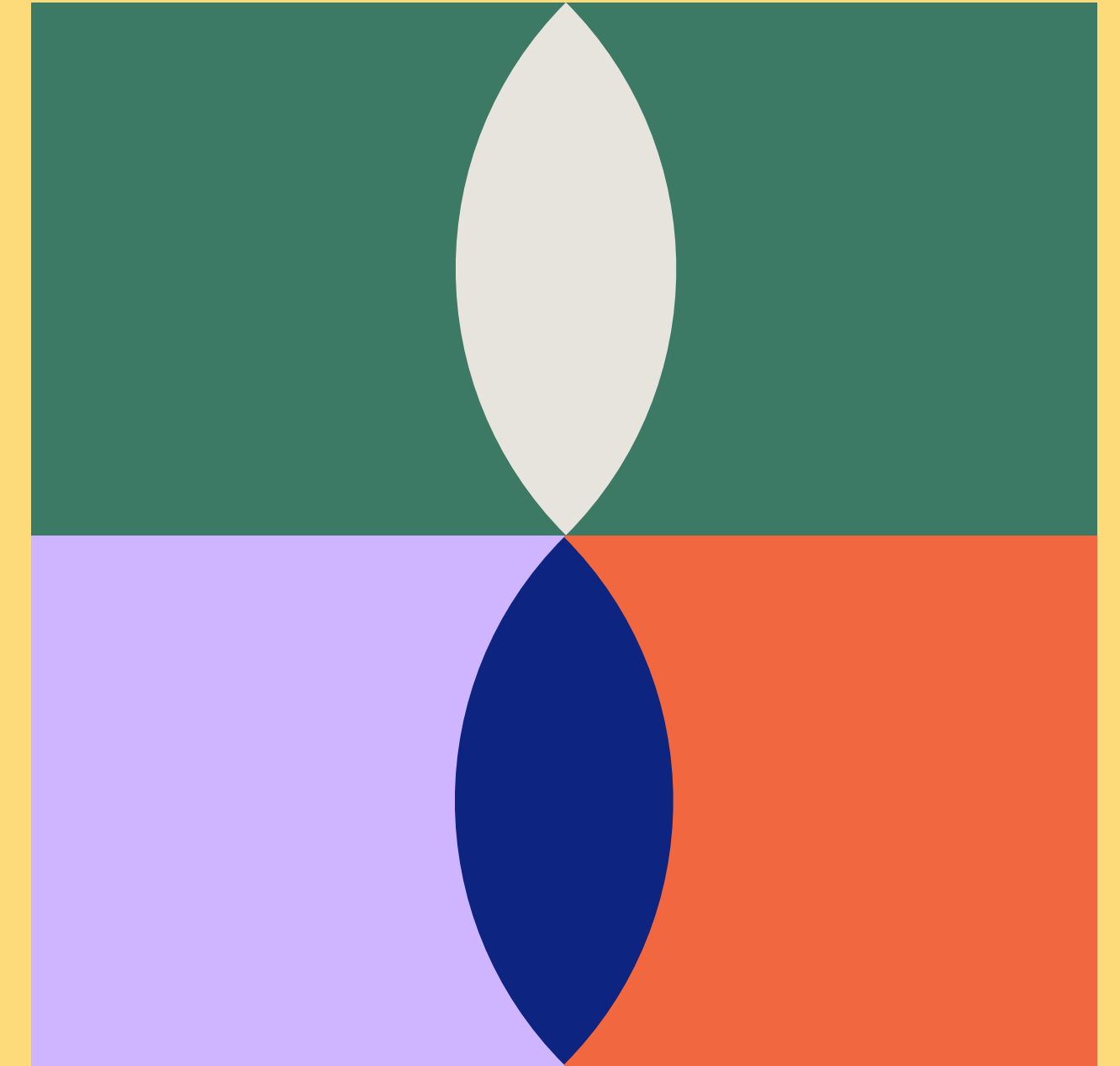
What's The Plan?

- We're here because we love CTFs
- We're also very ready for the end of the day celebration



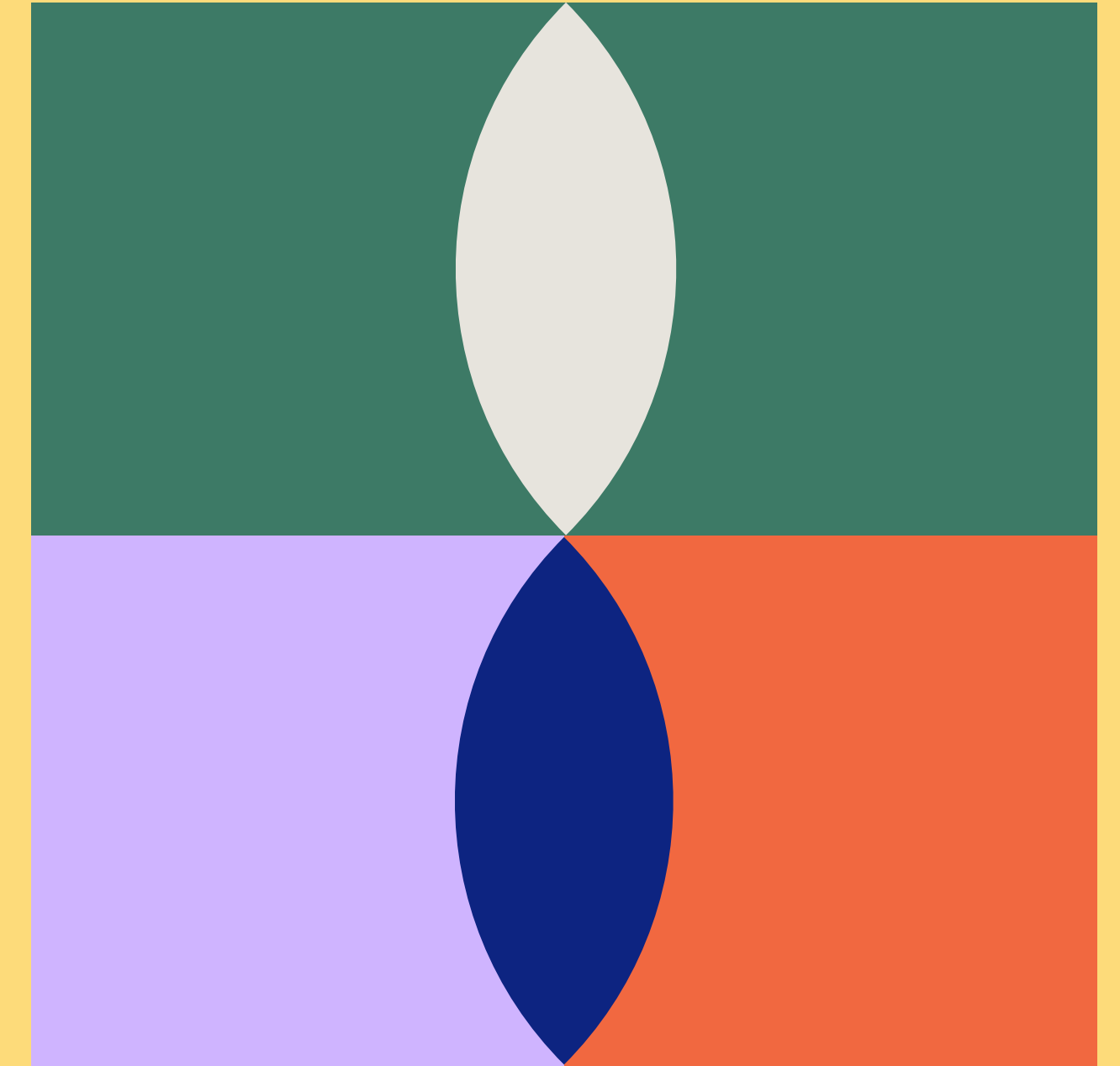
What's The Plan?

- We're here because we love CTFs
- We're also very ready for the end of the day celebration
- I want to talk about how I came into Cryptography via CTFs



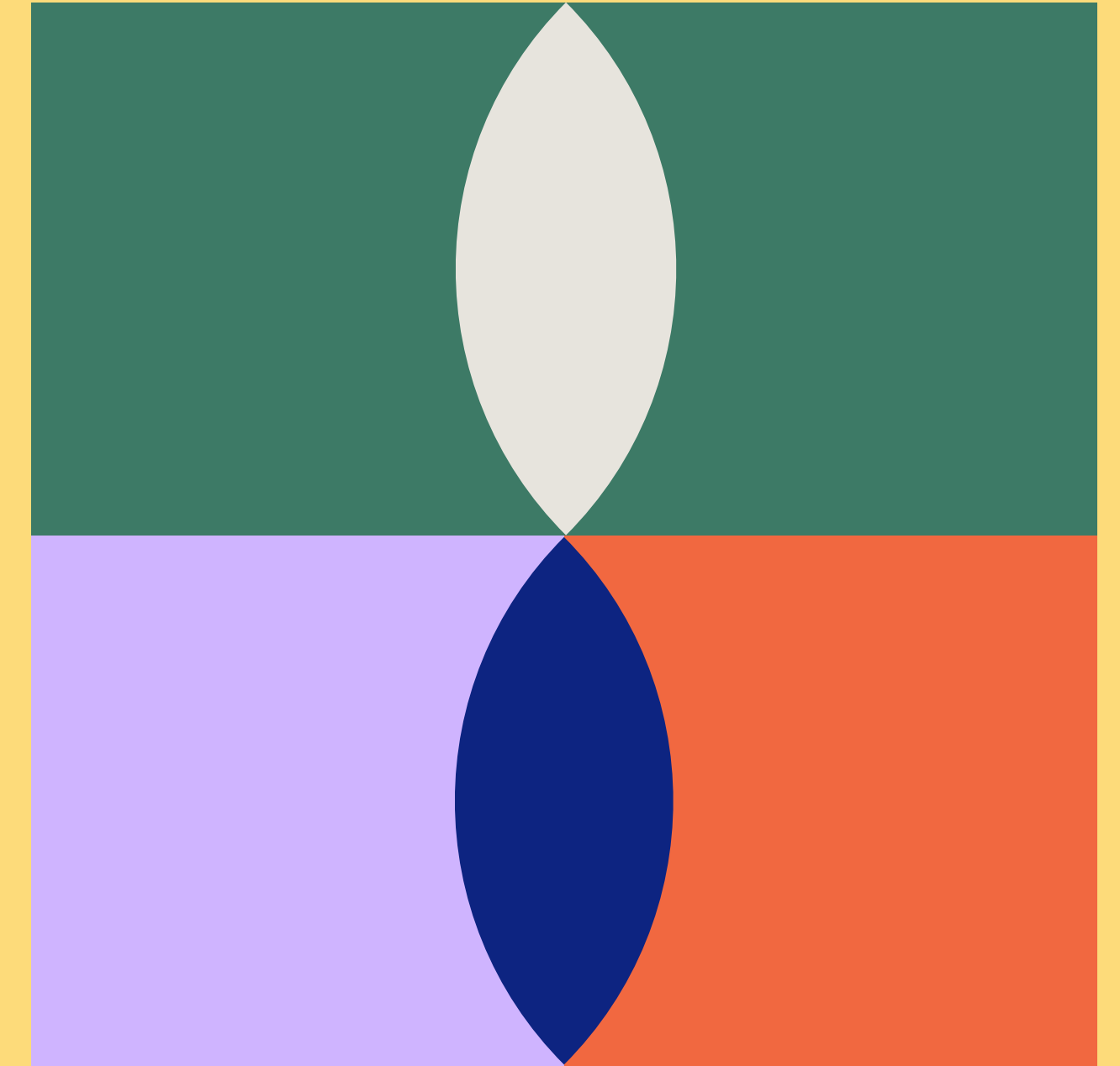
What's The Plan?

- We're here because we love CTFs
- We're also very ready for the end of the day celebration
- I want to talk about how I came into Cryptography via CTFs
- Discuss how what I do now is similar and distinct from our hobby



What's The Plan?

- We're here because we love CTFs
- We're also very ready for the end of the day celebration
- I want to talk about how I came into Cryptography via CTFs
- Discuss how what I do now is similar and distinct from our hobby
- Be available to guide folks interested in this direction



From Physics to Turin



From Physics to Turin

I started my PhD in 2016

From CTFs to Cryptography

From Physics to Turin

I started my PhD in 2016

I started CTFs in 2019

From CTFs to Cryptography

From Physics to Turin

I started my PhD in 2016

I started CTFs in 2019

I co-created
CryptoHack in 2020

From CTFs to Cryptography

From Physics to Turin

I started my PhD in 2016

I started CTFs in 2019

I co-created
CryptoHack in 2020

I joined NCC Group in 2021

From CTFs to Cryptography

From Physics to Turin

I started my PhD in 2016

I started CTFs in 2019

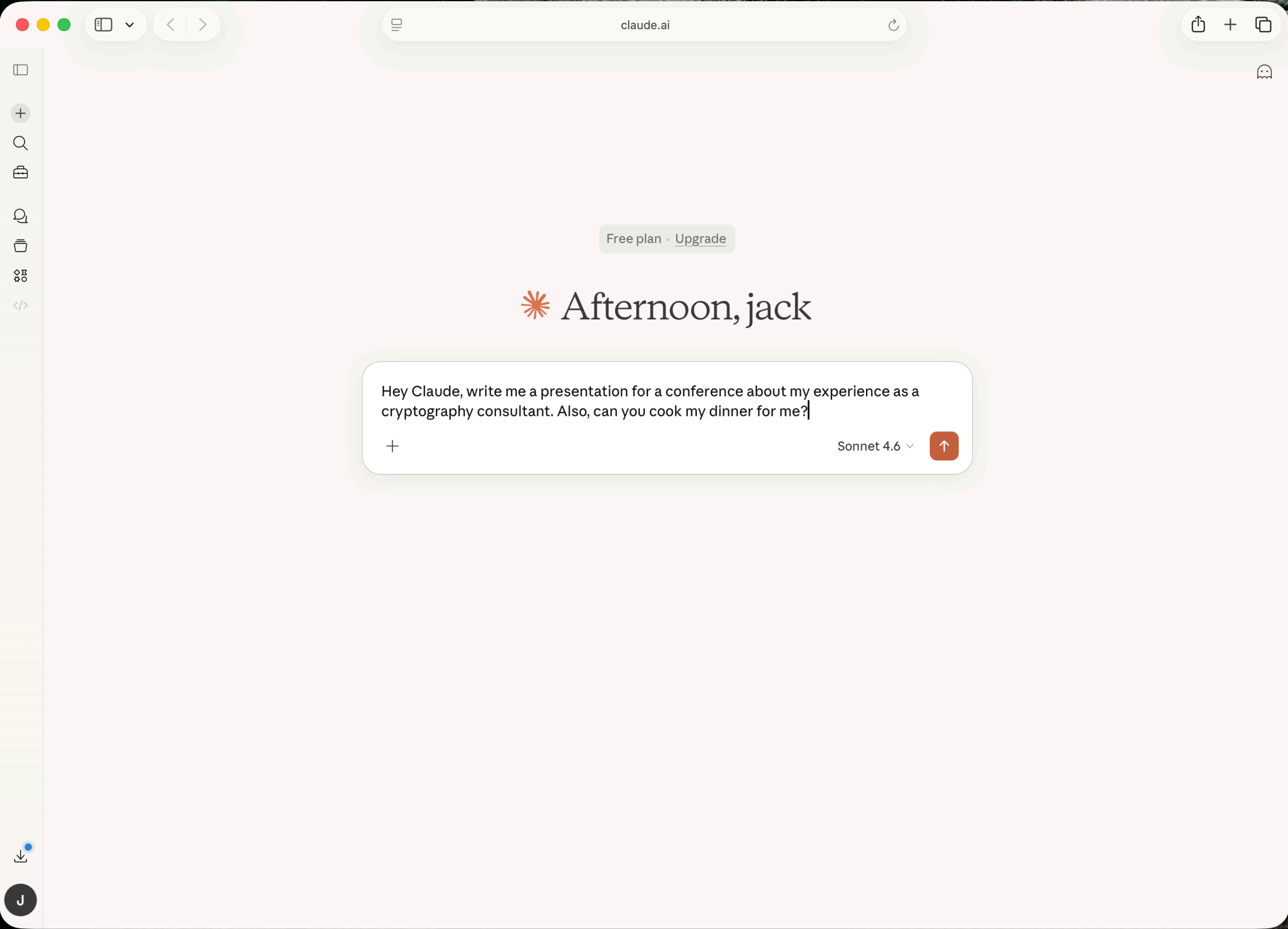
I co-created
CryptoHack in 2020

I joined NCC Group in 2021

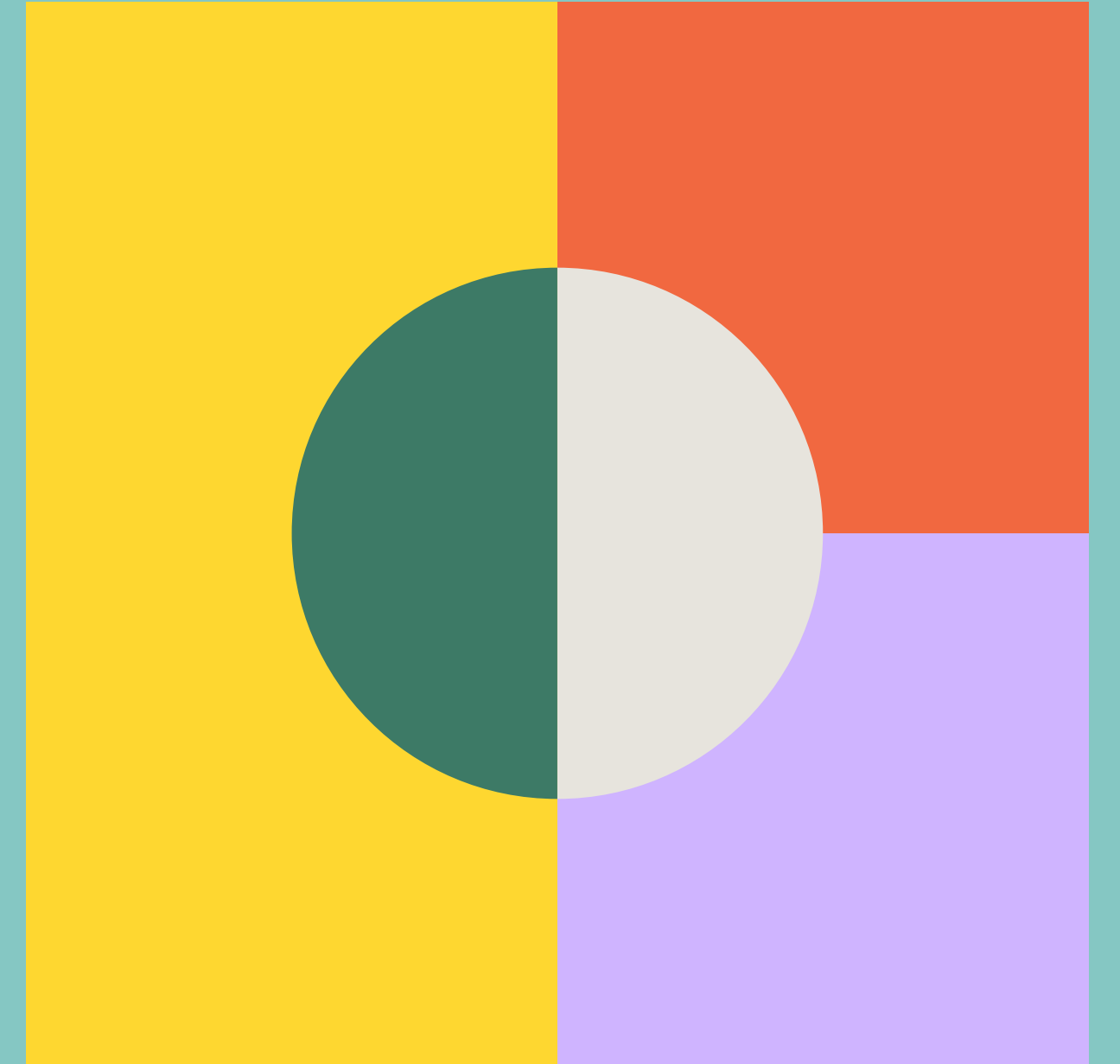
I flew to Turin yesterday

From CTFs to Cryptography



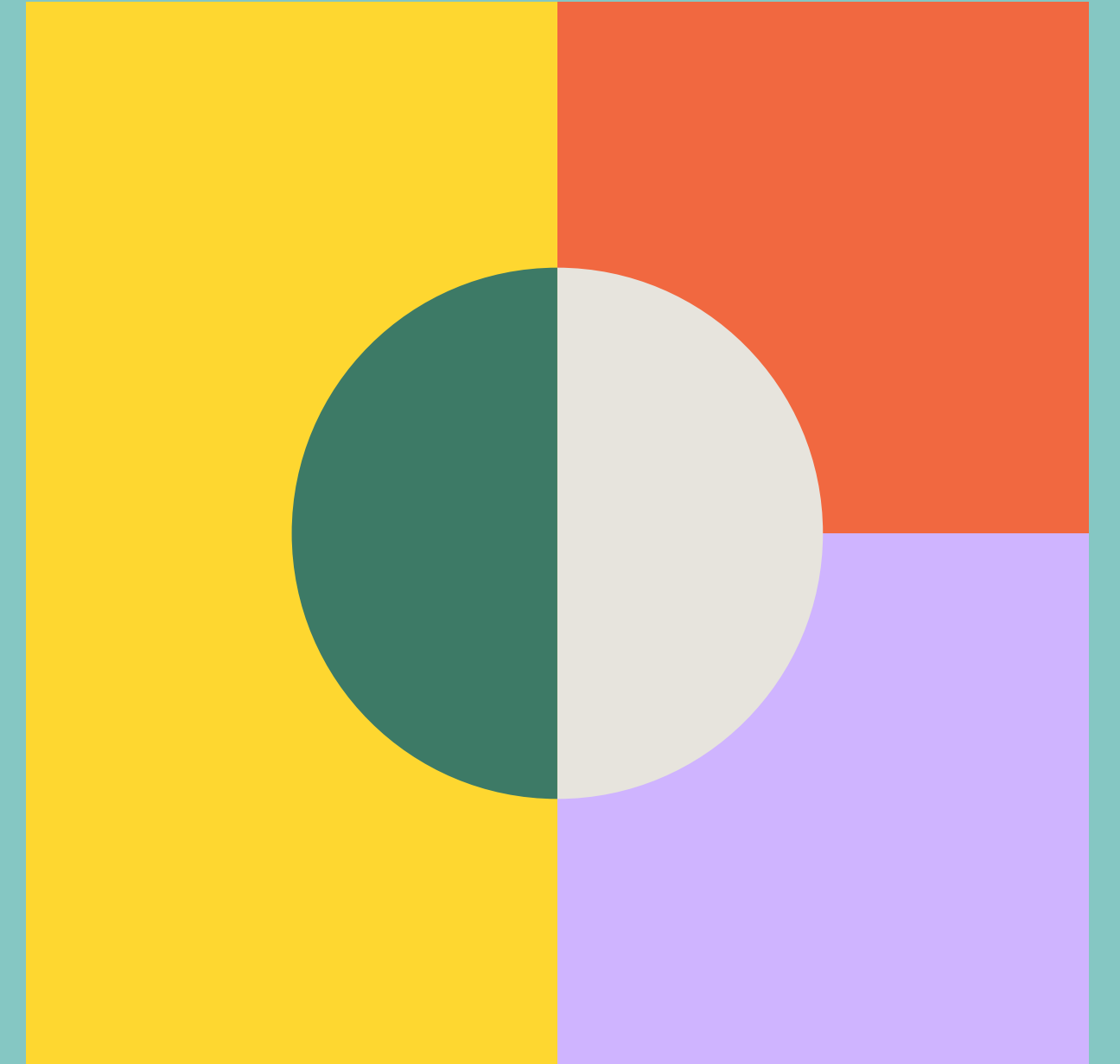


Capture the Bug



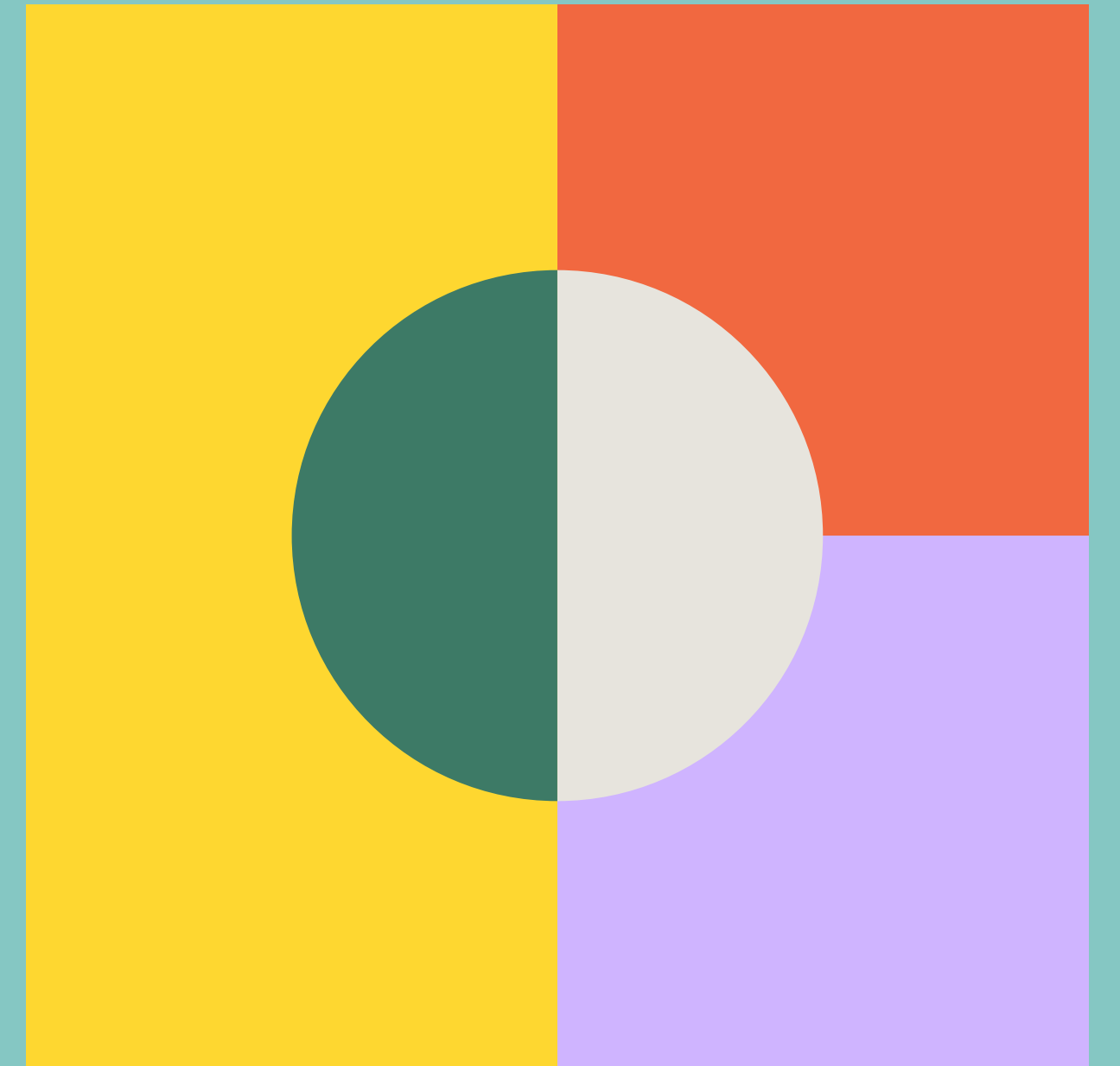
Capture the Bug

- I love solving cryptography puzzles.



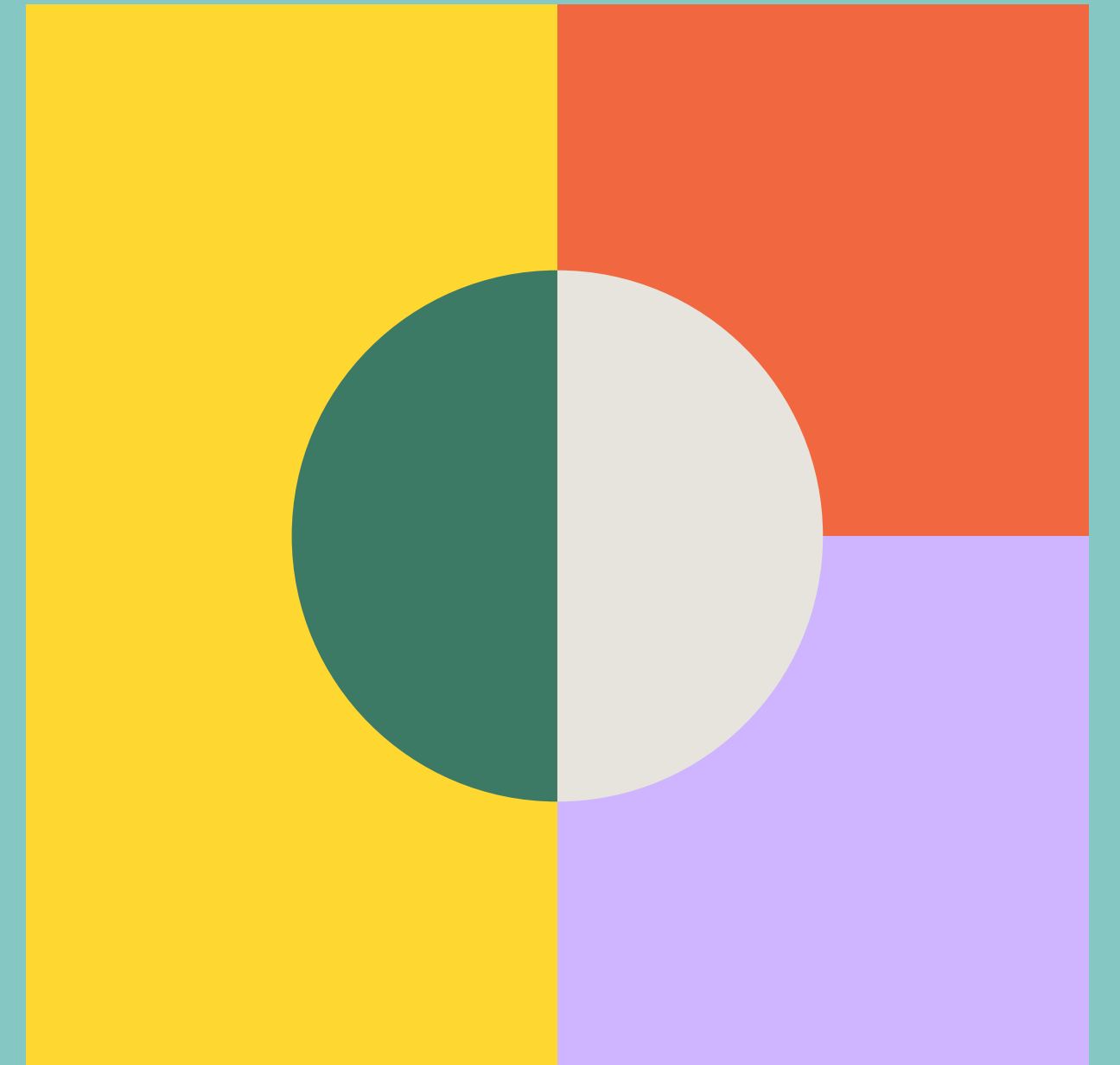
Capture the Bug

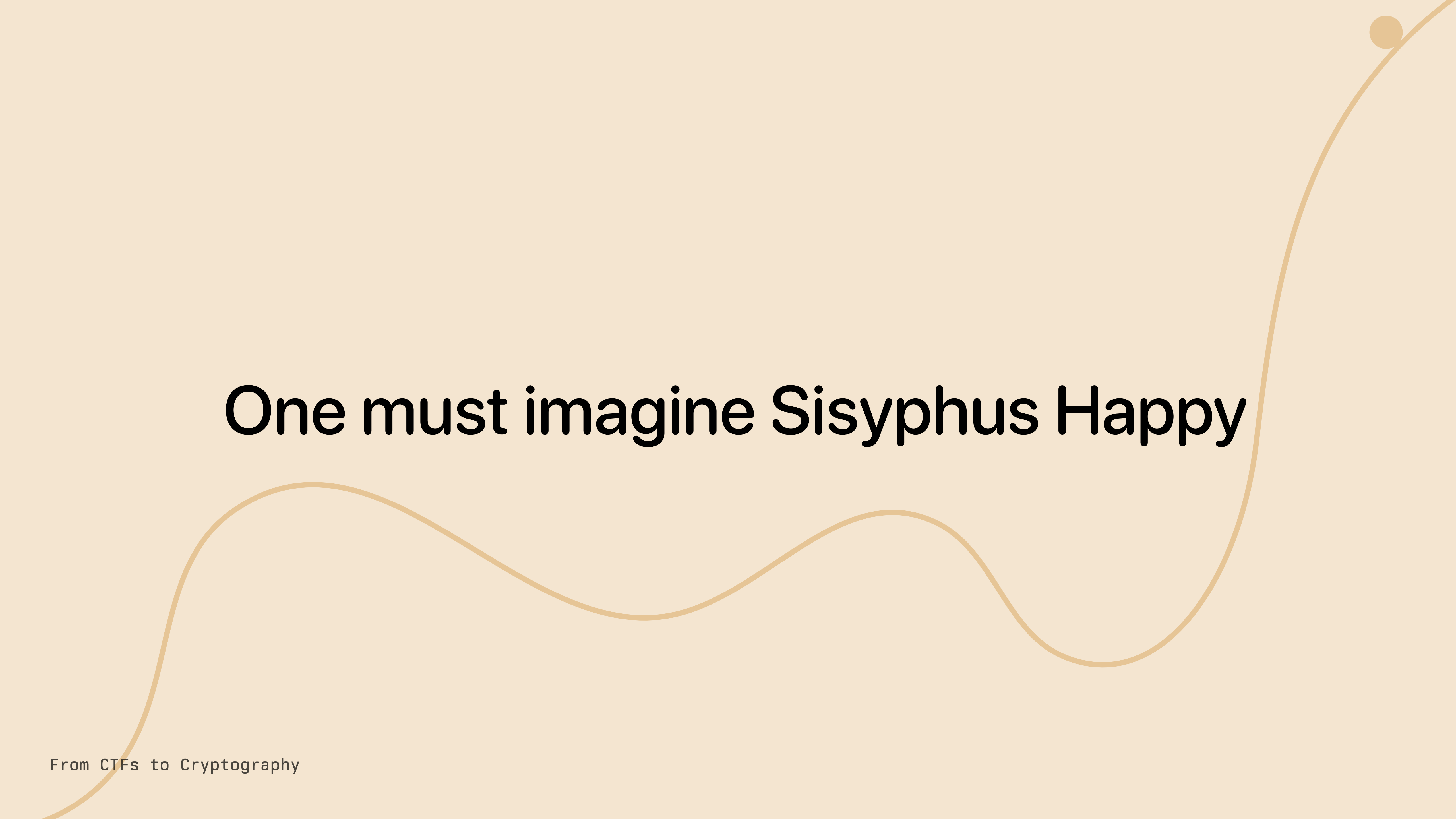
- I love solving cryptography puzzles.
- A challenge from PlaidCTF 2020 has directed the past 5 years of my research!



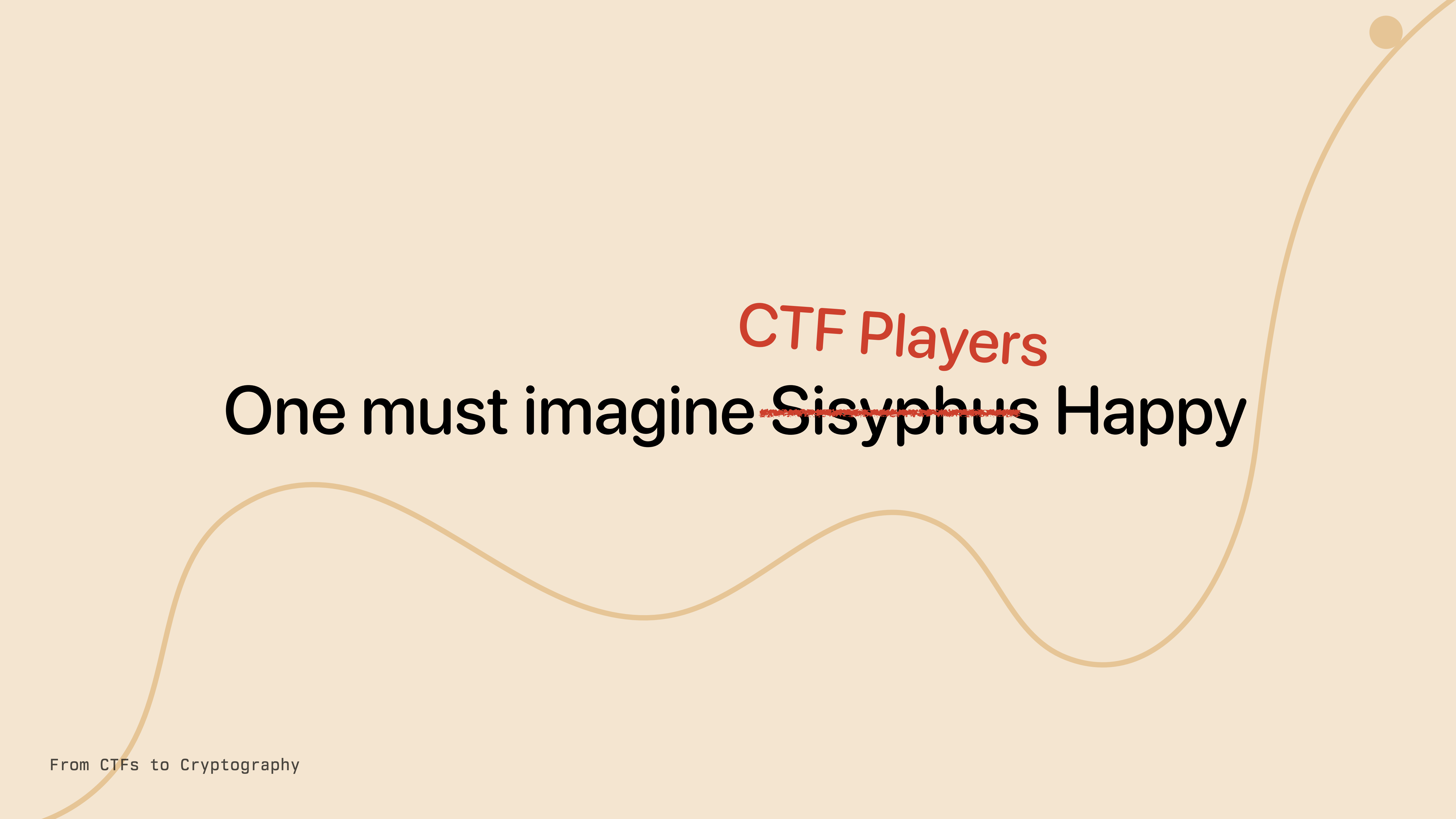
Capture the Bug

- I love solving cryptography puzzles.
- A challenge from PlaidCTF 2020 has directed the past 5 years of my research!
- Can I make searching for and then implementing mathematical attacks a job?



A decorative orange line starts from the bottom left, curves upwards, then wavyly across the middle, and finally curves sharply upwards to the top right, ending with a solid orange dot.

One must imagine Sisyphus Happy

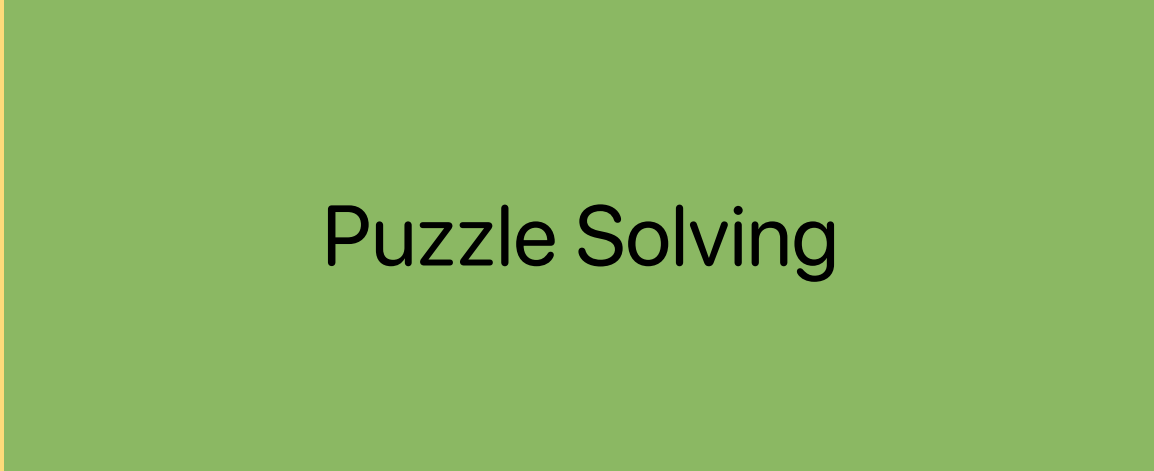
A decorative orange wavy line starts from the bottom left, curves upwards, then downwards, then upwards again, ending at the top right where it meets a small orange circle.

CTF Players

One must imagine ~~Sisyphus~~ Happy

The CTF Experience

The CTF Experience



Puzzle Solving

The CTF Experience

Puzzle Solving

Sharing Writeups

The CTF Experience

Puzzle Solving

Learning in Teams

Sharing Writeups

The CTF Experience

Puzzle Solving

Learning in Teams

Sharing Writeups

Winning Competitions

The CTF Experience

Puzzle Solving

Learning in Teams

Sharing Writeups

Winning Competitions

Racing for First Blood

The CTF Experience

Puzzle Solving

Learning in Teams

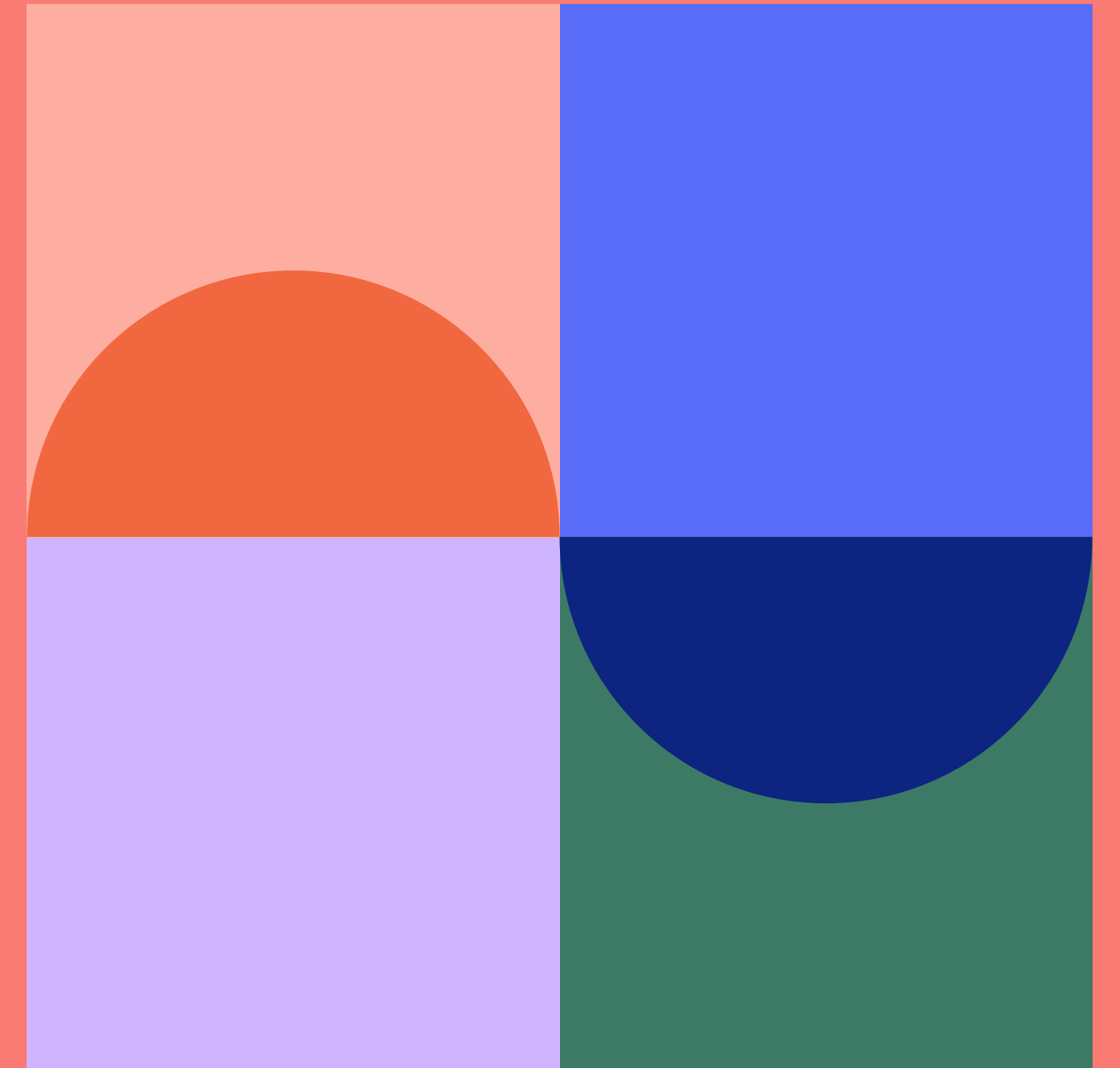
Sharing Writeups

Winning Competitions

Travelling the World

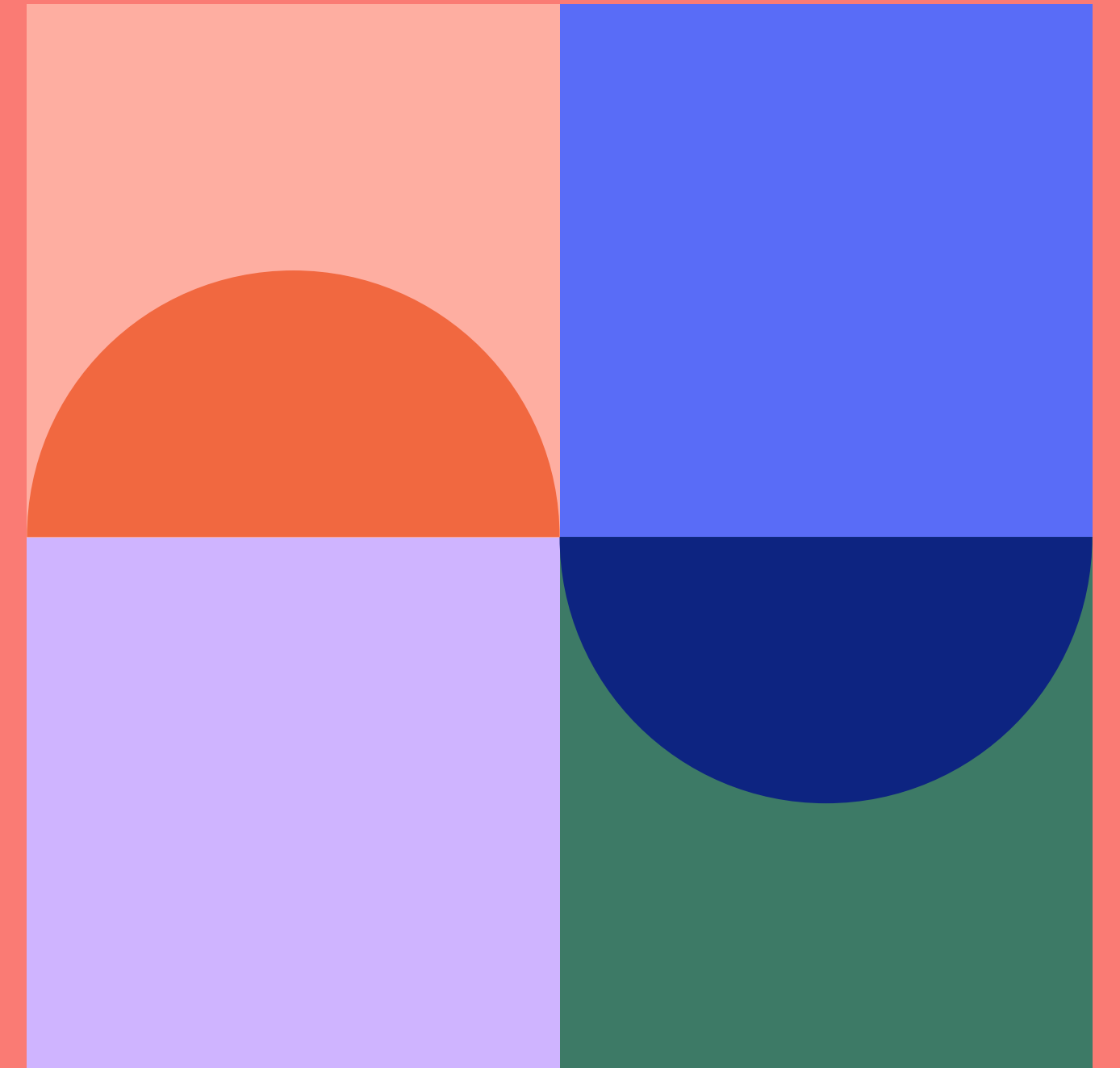
Racing for First Blood

Cryptographic Services



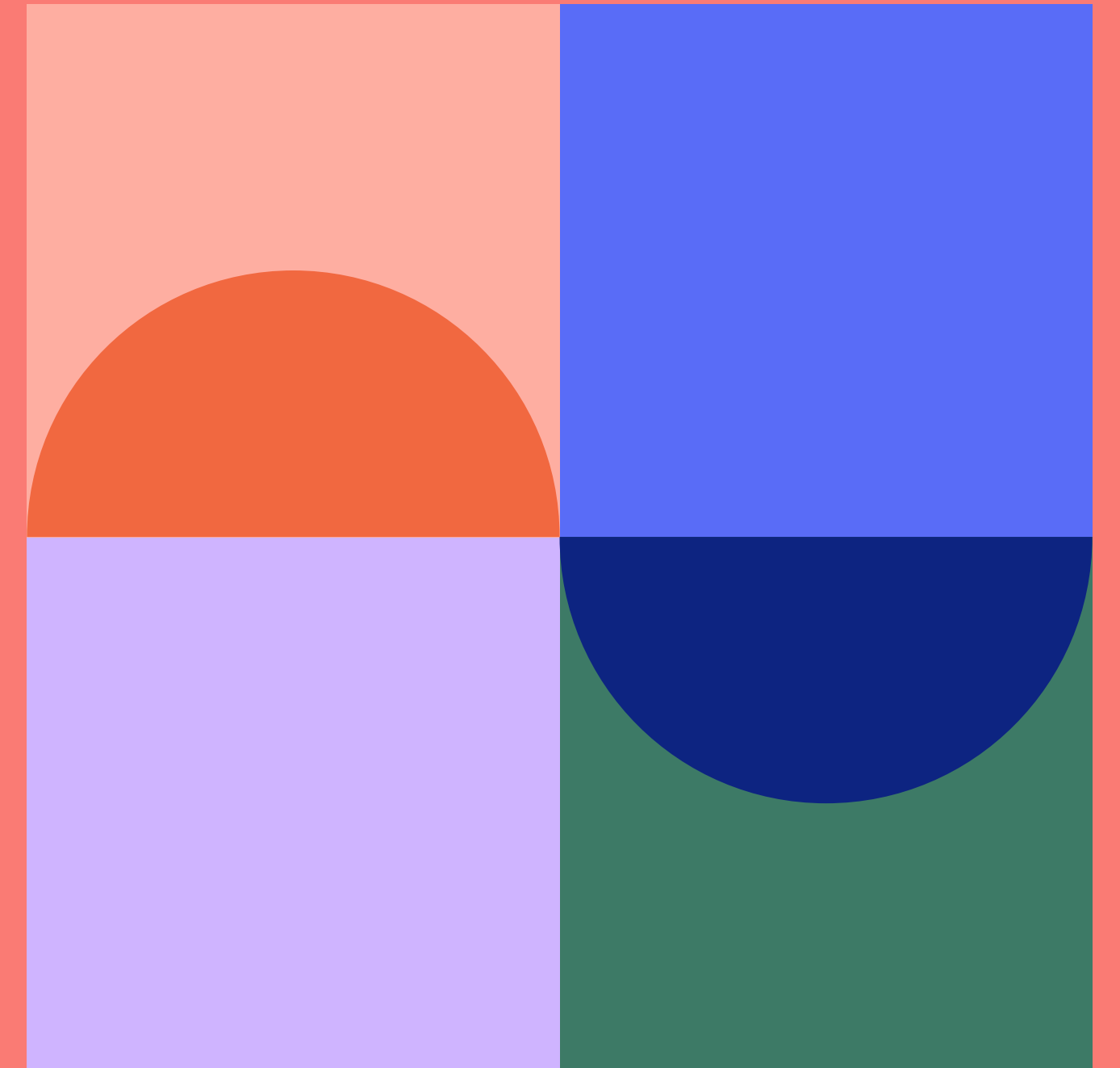
Cryptographic Services

- Work comes in distinct forms



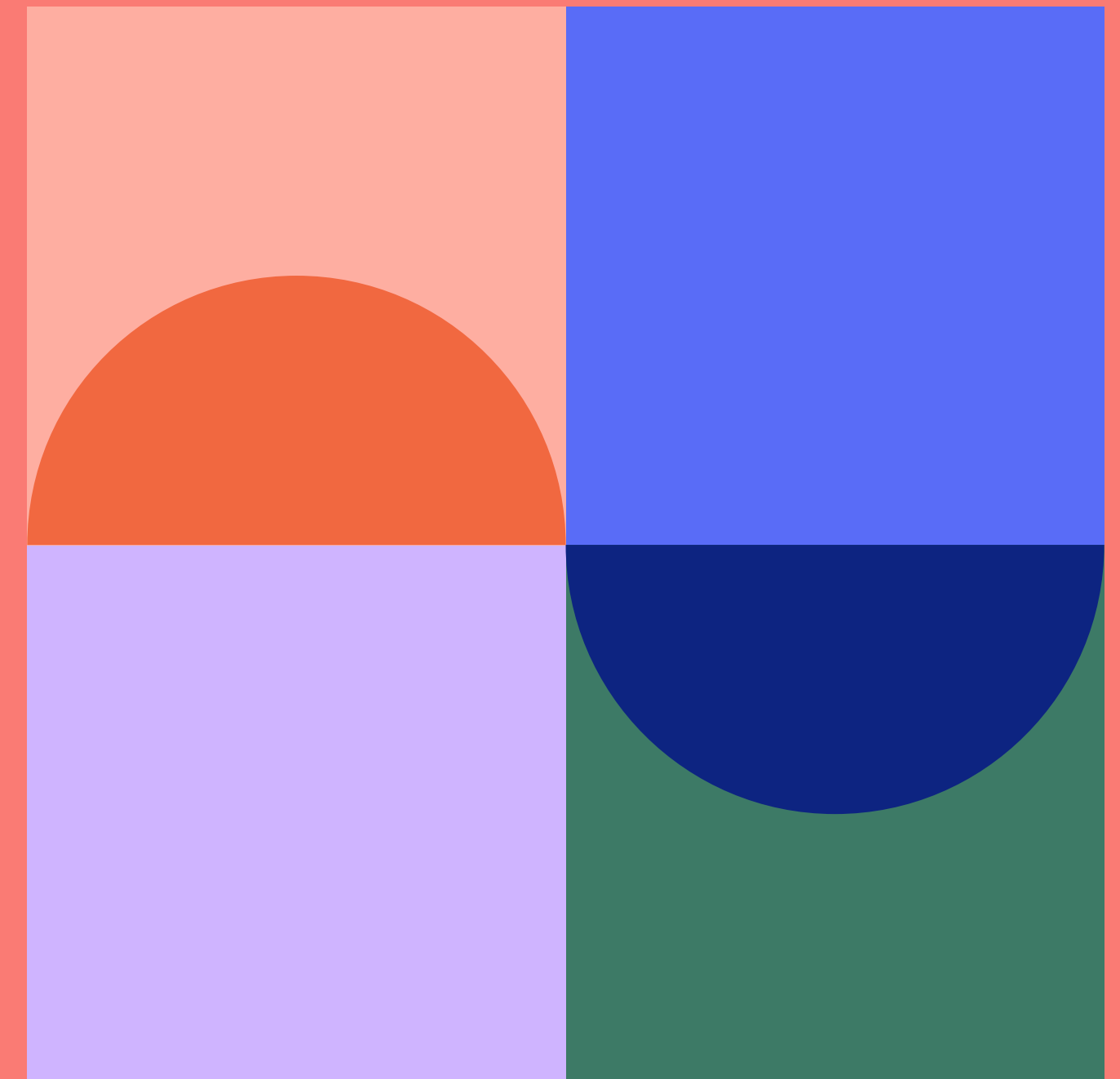
Cryptographic Services

- Work comes in distinct forms
 - Architecture / Design Reviews



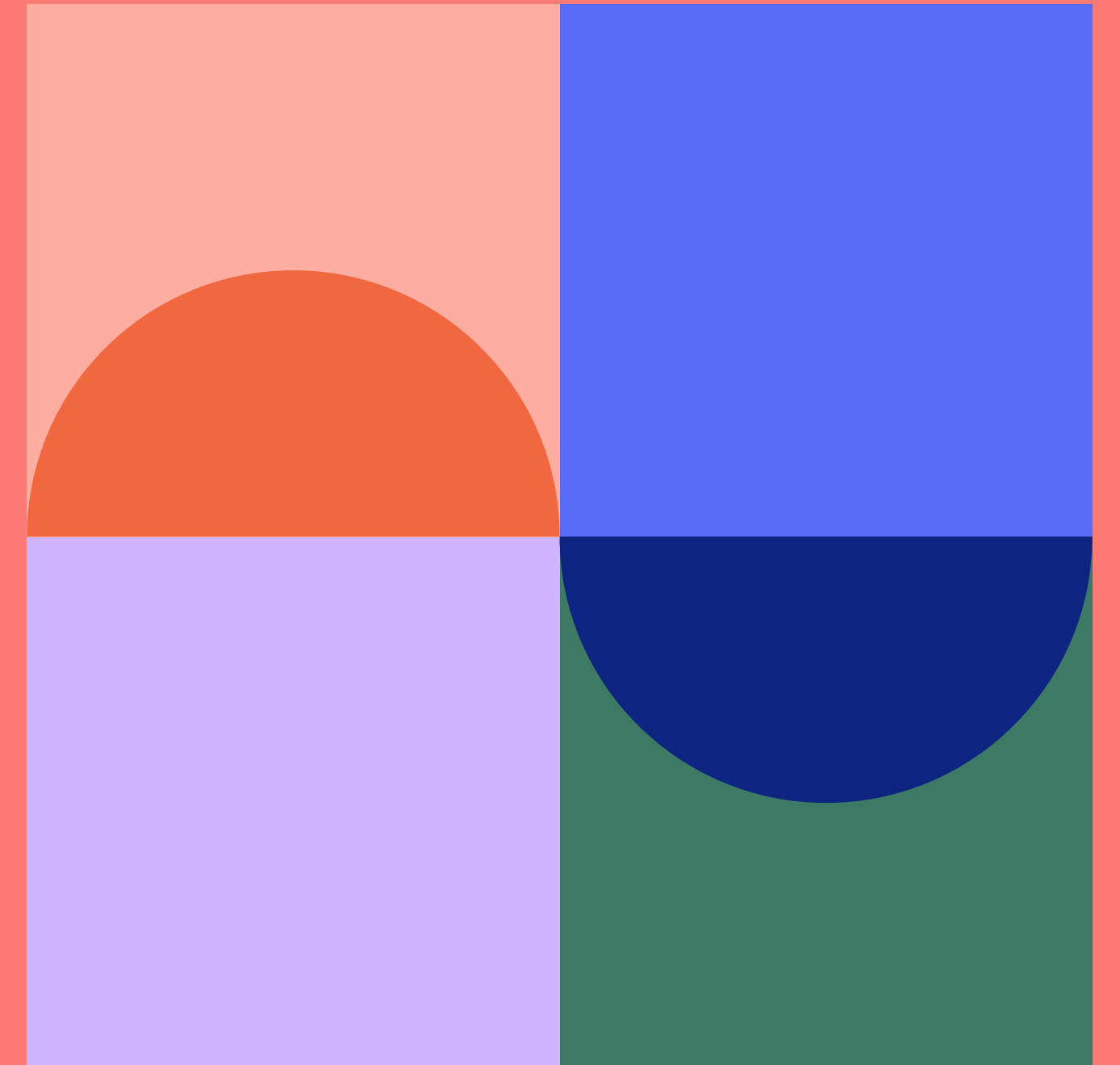
Cryptographic Services

- Work comes in distinct forms
 - Architecture / Design Reviews
 - Core primitive implementation reviews

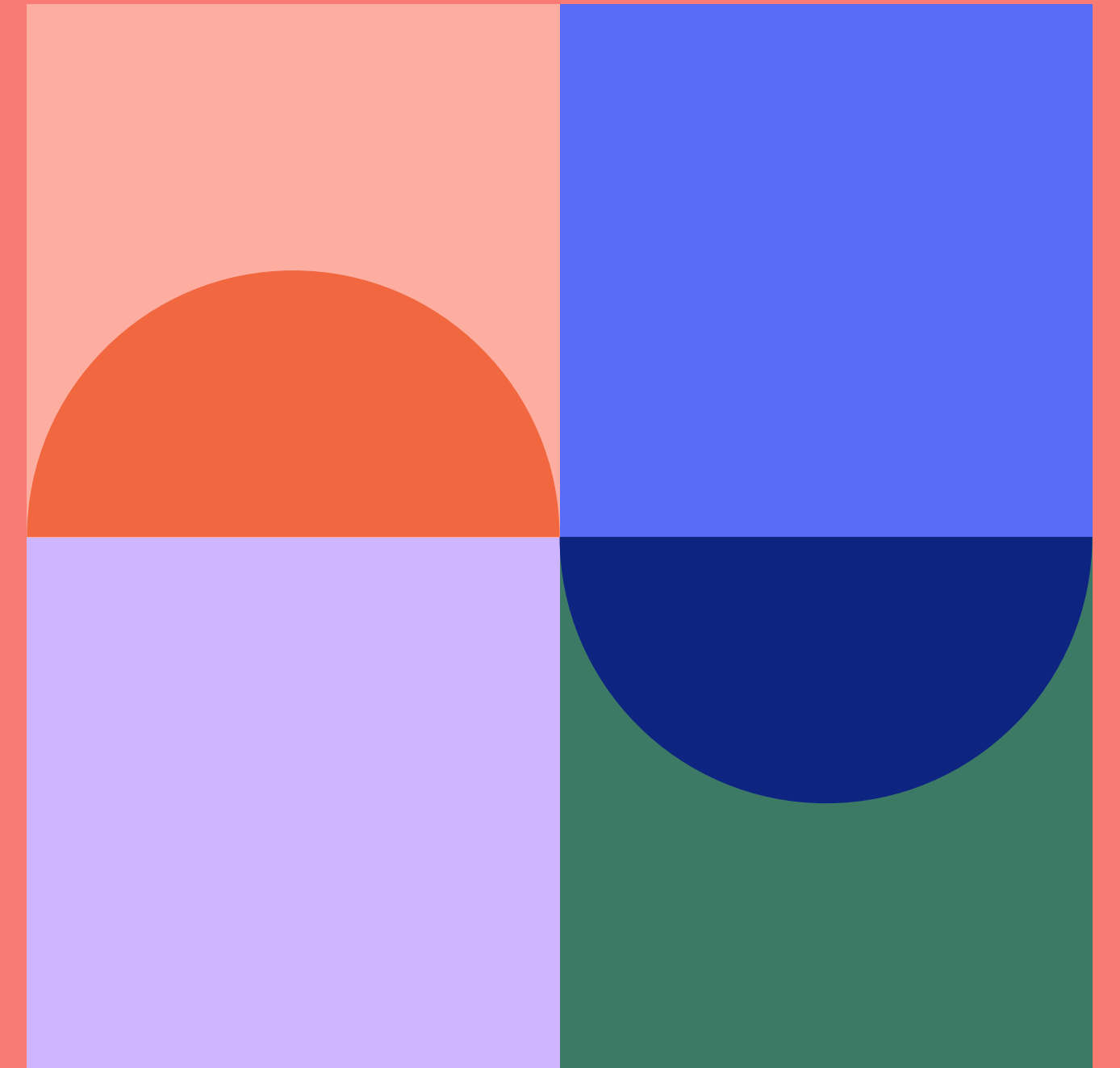


Cryptographic Services

- Work comes in distinct forms
 - Architecture / Design Reviews
 - Core primitive implementation reviews
 - Cryptographic application reviews

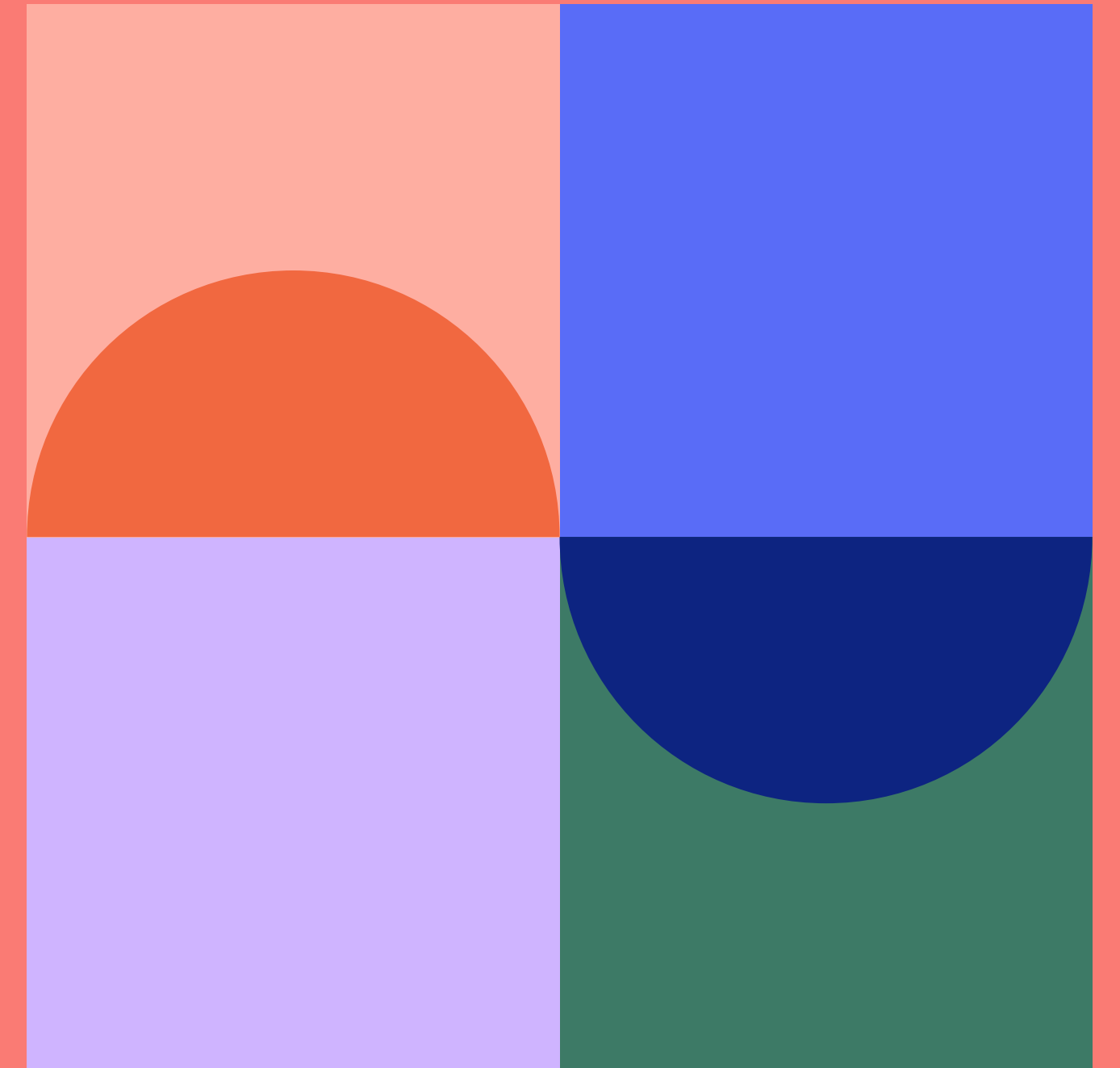


Cryptographic Services



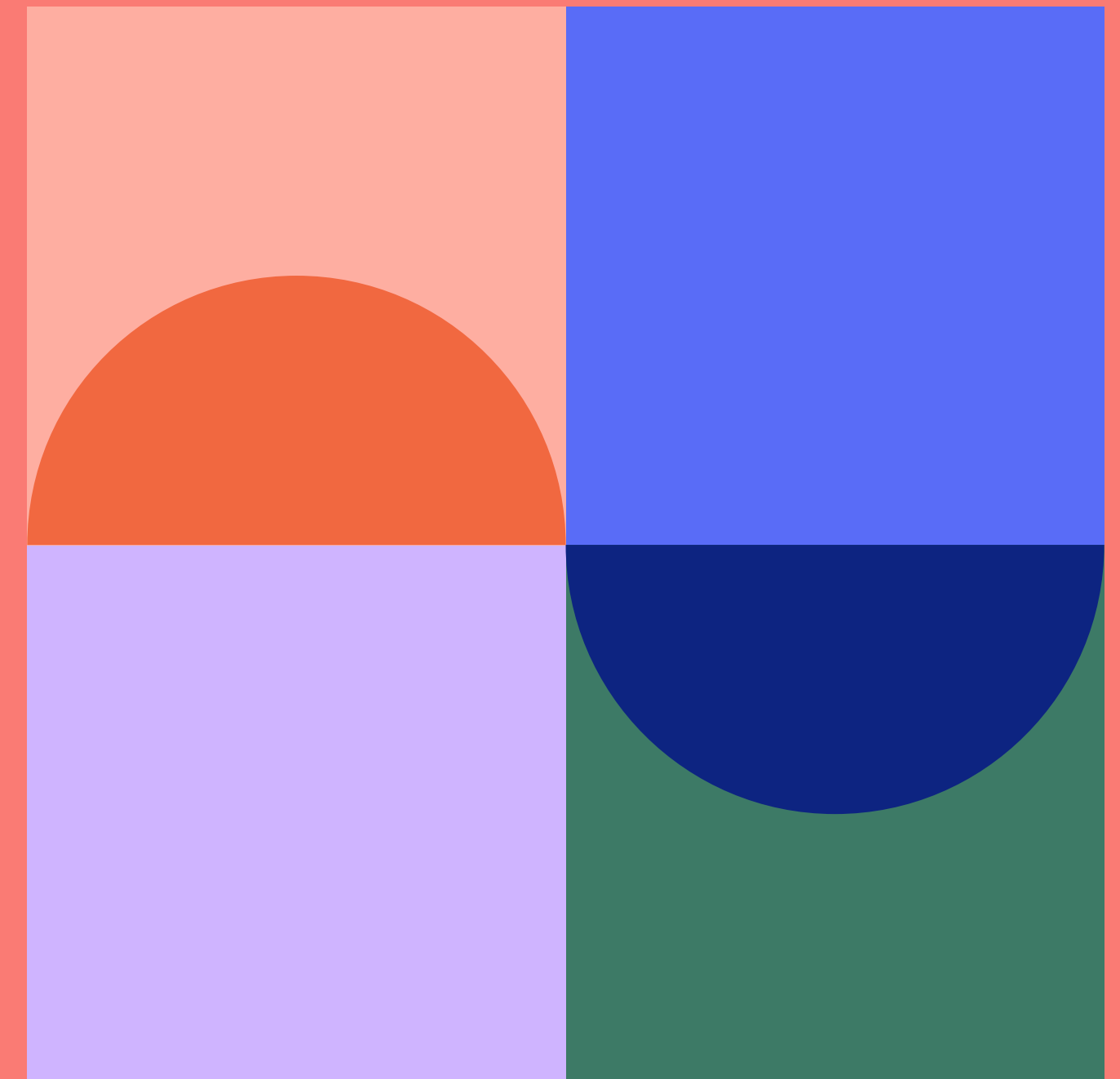
Cryptographic Services

- Typically a project is 1-2 weeks long



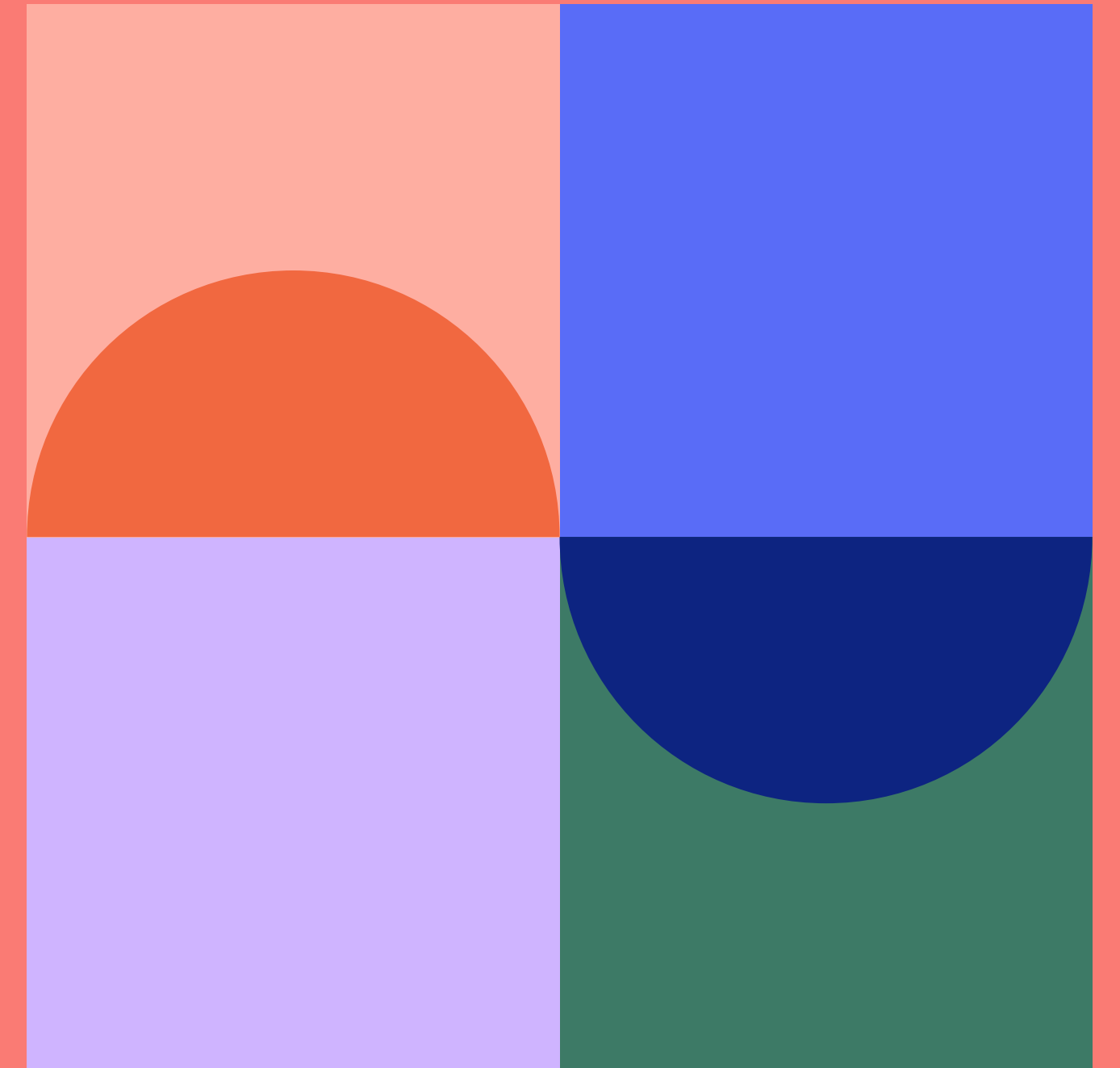
Cryptographic Services

- Typically a project is 1-2 weeks long
- Work in small groups, sometimes collaborating with other specialists



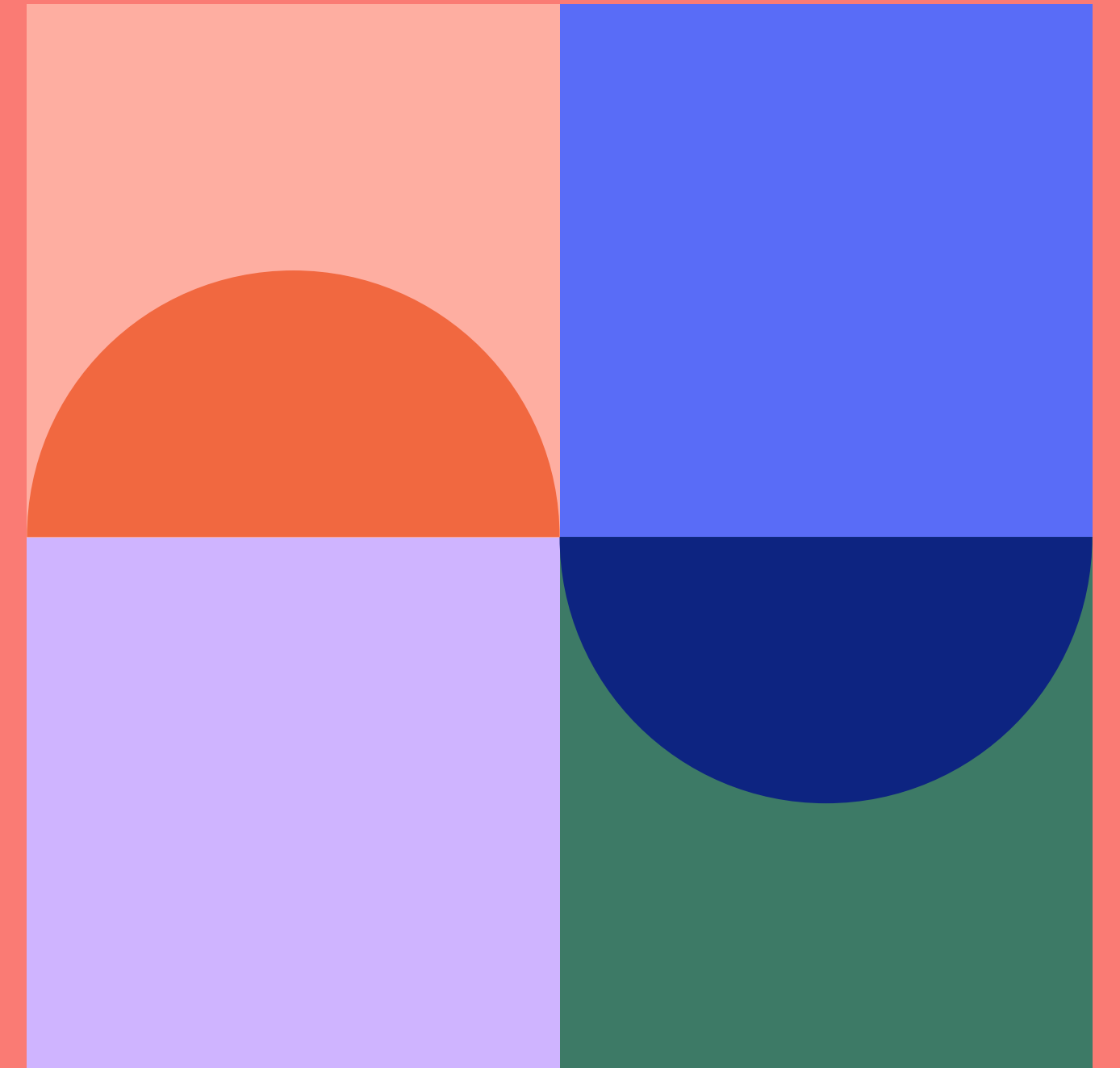
Cryptographic Services

- Typically a project is 1-2 weeks long
- Work in small groups, sometimes collaborating with other specialists
- A project starts with a kick-off chat



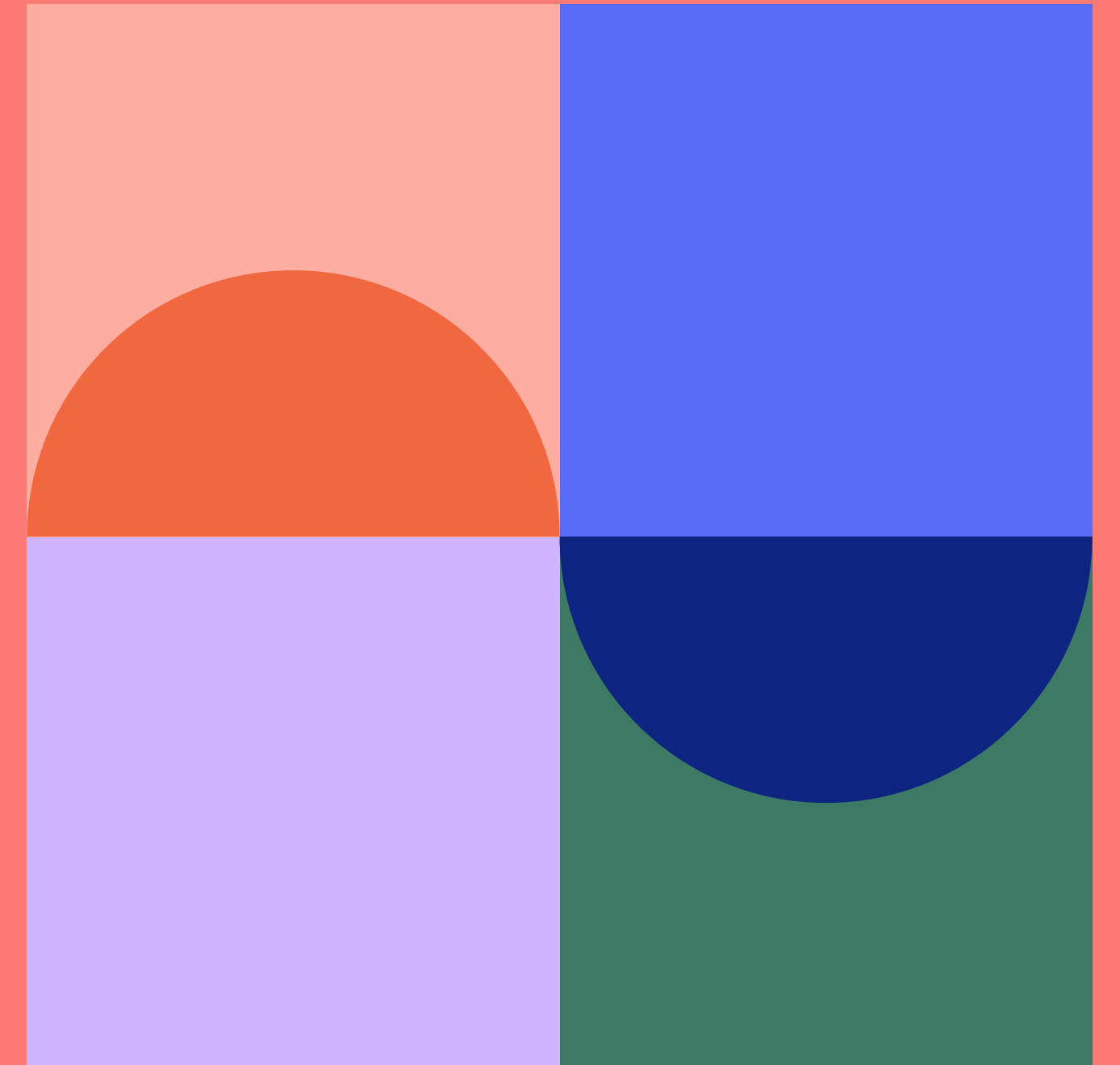
Cryptographic Services

- Typically a project is 1-2 weeks long
- Work in small groups, sometimes collaborating with other specialists
- A project starts with a kick-off chat
- Each week we have a catch up on the current progress



Cryptographic Services

- Typically a project is 1-2 weeks long
- Work in small groups, sometimes collaborating with other specialists
- A project starts with a kick-off chat
- Each week we have a catch up on the current progress
- A finished project ends with a formal report outlining problems and potential solutions



Real Real World CTF

Real Real World CTF

API Misuse

AES-GCM Nonce Reuse

Real Real World CTF

API Misuse

AES-GCM Nonce Reuse

Input Validation

Signature Verification Bypass

Real Real World CTF

API Misuse

AES-GCM Nonce Reuse

Side Channel Attacks

Variable time byte
comparison

Input Validation

Signature Verification Bypass

Real Real World CTF

API Misuse

AES-GCM Nonce Reuse

Side Channel Attacks

Variable time byte
comparison

Input Validation

Signature Verification Bypass

Mismatching Theory

Weak Fiat Shamir
Transforms

Real Real World CTF

API Misuse

AES-GCM Nonce Reuse

Side Channel Attacks

Variable time byte
comparison

Input Validation

Signature Verification Bypass

Mismatching Theory

Weak Fiat Shamir
Transforms

Implementation Errors

goto fail; goto fail;

Real Real World CTF

API Misuse

AES-GCM Nonce Reuse

Side Channel Attacks

Variable time byte
comparison

Input Validation

Signature Verification Bypass

Mismatching Theory

Weak Fiat Shamir
Transforms

Weak Randomness

Bias in random
number generation

Implementation Errors

goto fail; goto fail;

goto.c Sign In

goto.c

goto.c > OSStatus SSLVerifySignedServerKeyExchange()

```
45 hashOut.data = hashes + SSL_MD5_DIGEST_LEN;
46 hashOut.length = SSL_SHA1_DIGEST_LEN;
47 if ((err = SSLFreeBuffer(&hashCtx)) != 0)
48     goto fail;
49
50 if ((err = ReadyHash(&SSLHashSHA1, &hashCtx)) != 0)
51     goto fail;
52 if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
53     goto fail;
54 if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
55     goto fail;
56 if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
57     goto fail;
58     goto fail;
59 if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
60     goto fail;
61
62 err = sslRawVerify(ctx,
63                   ctx->peerPubKey,
64                   dataToSign,      /* plaintext */
65                   dataToSignLen,   /* plaintext length */
66                   signature,
67                   signatureLen);
```

45:2 C Z»

goto.c Sign In

goto.c

goto.c > OSStatus SSLVerifySignedServerKeyExchange()

```
45 hashOut.data = hashes + SSL_MD5_DIGEST_LEN;
46 hashOut.length = SSL_SHA1_DIGEST_LEN;
47 if ((err = SSLFreeBuffer(&hashCtx)) != 0)
48     goto fail;
49
50 if ((err = ReadyHash(&SSLHashSHA1, &hashCtx)) != 0)
51     goto fail;
52 if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
53     goto fail;
54 if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
55     goto fail;
56 if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
57     goto fail;
58     goto fail;
59 if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
60     goto fail;
61
62 err = sslRawVerify(ctx,
63                   ctx->peerPubKey,
64                   dataToSign,      /* plaintext */
65                   dataToSignLen,   /* plaintext length */
66                   signature,
67                   signatureLen);
```

45:2 C Z»

```
def gen_keypair():  
    """  
    Generate an RSA keypair  
    """  
    e, d = 1, -1  
    while d == -1:  
        p = getPrime(1024)  
        q = getPrime(1024)  
        phi = (p - 1) * (q - 1)  
        d = inverse(e, phi)  
    n = p * q  
    return (N, e), (N, e, d)
```




1 file changed +1 -1 lines changed

Search within code



salt/crypt.py

+1 -1



@@ -47,7 +47,7 @@ def gen_keys(keydir, keyname, keysize, user=None):

47 priv = '{0}.pem'.format(base)

48 pub = '{0}.pub'.format(base)

49

50 - gen = RSA.gen_key(keysize, 1, callback=lambda x, y, z: None)

51 cumask = os.umask(191)

52 gen.save_key(priv, None)

53 os.umask(cumask)



47 priv = '{0}.pem'.format(base)

48 pub = '{0}.pub'.format(base)

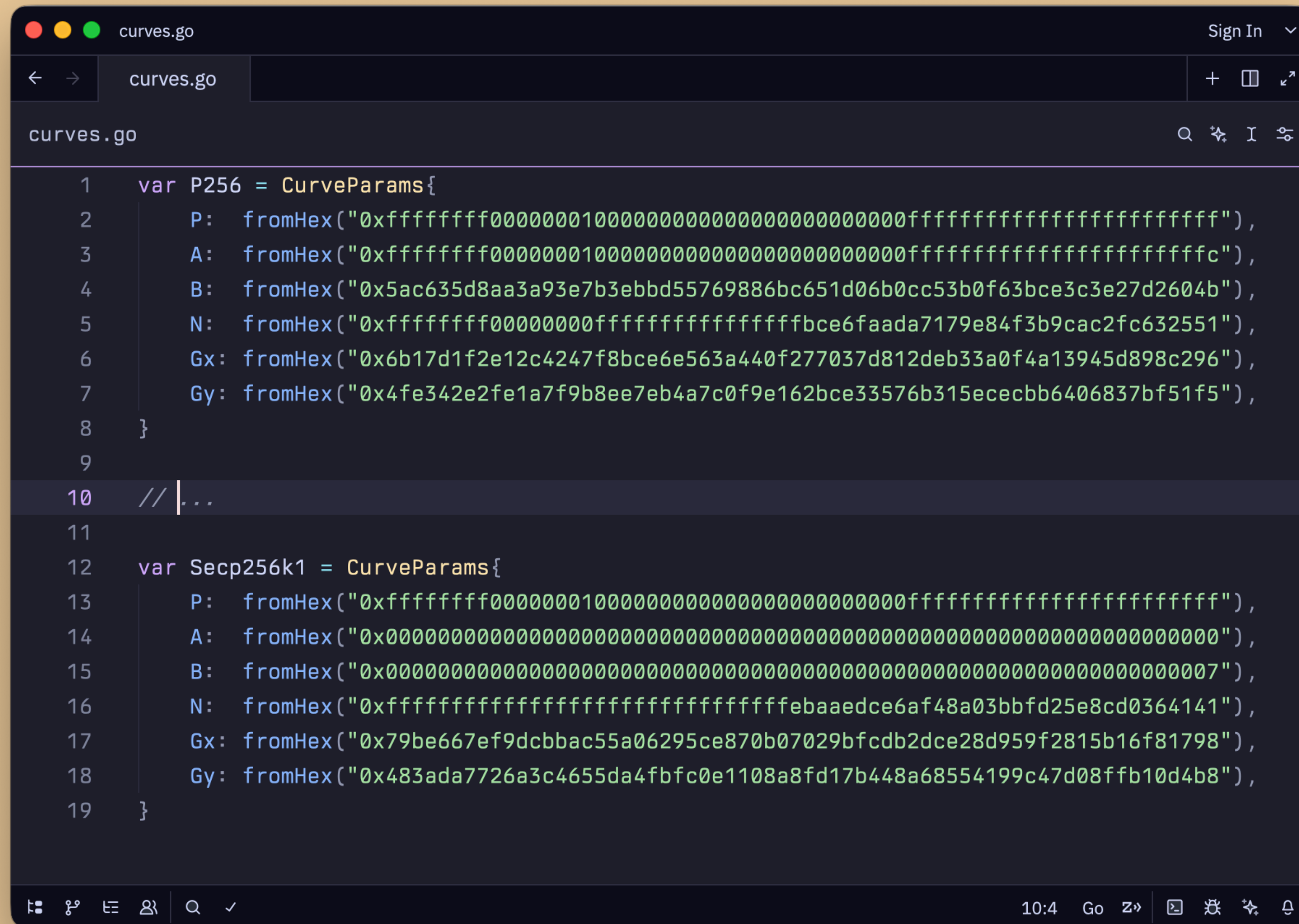
49

50 + gen = RSA.gen_key(keysize, 65537, callback=lambda x, y, z: None)

51 cumask = os.umask(191)

52 gen.save_key(priv, None)

53 os.umask(cumask)



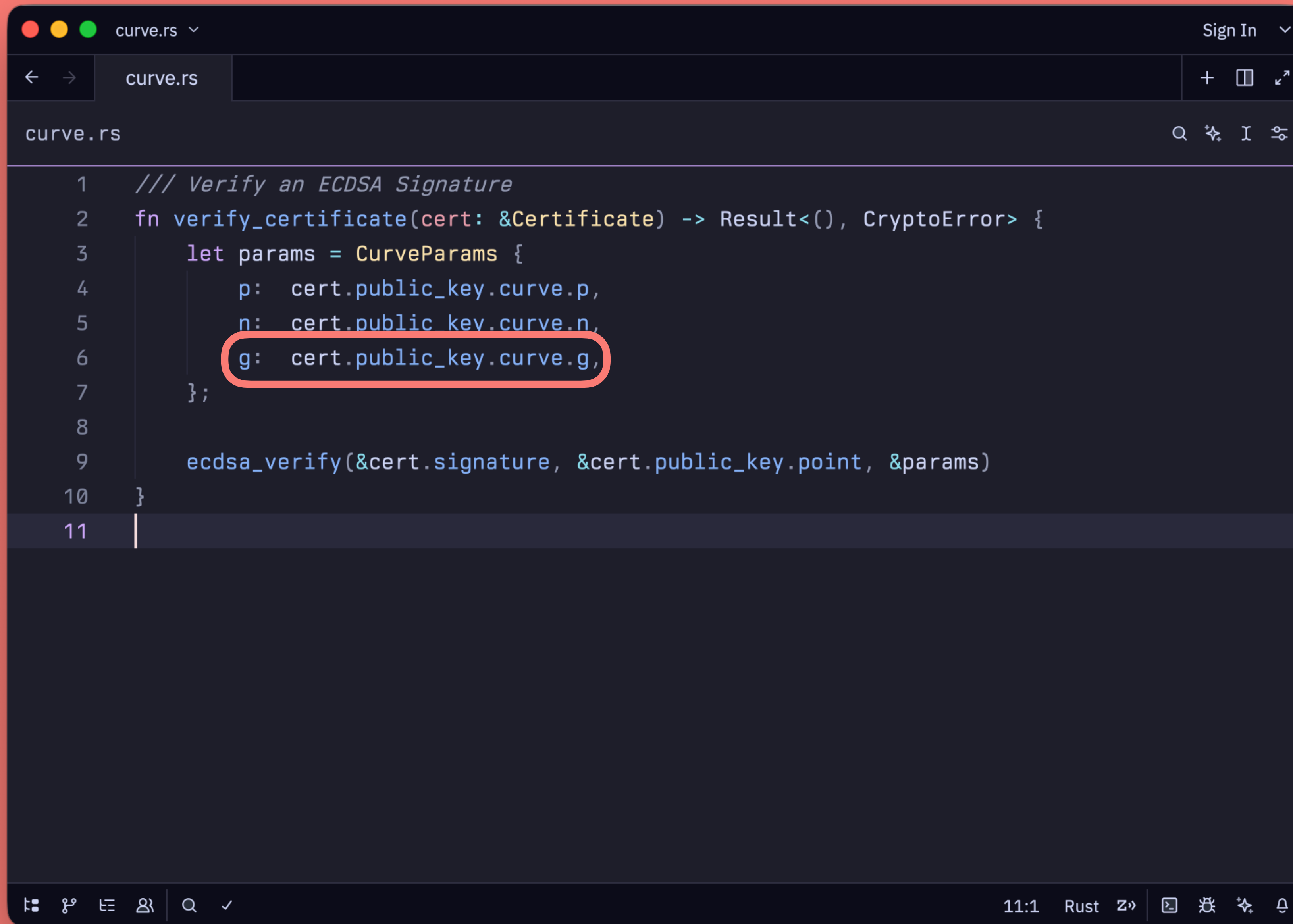
curve.rs

curve.rs

curve.rs

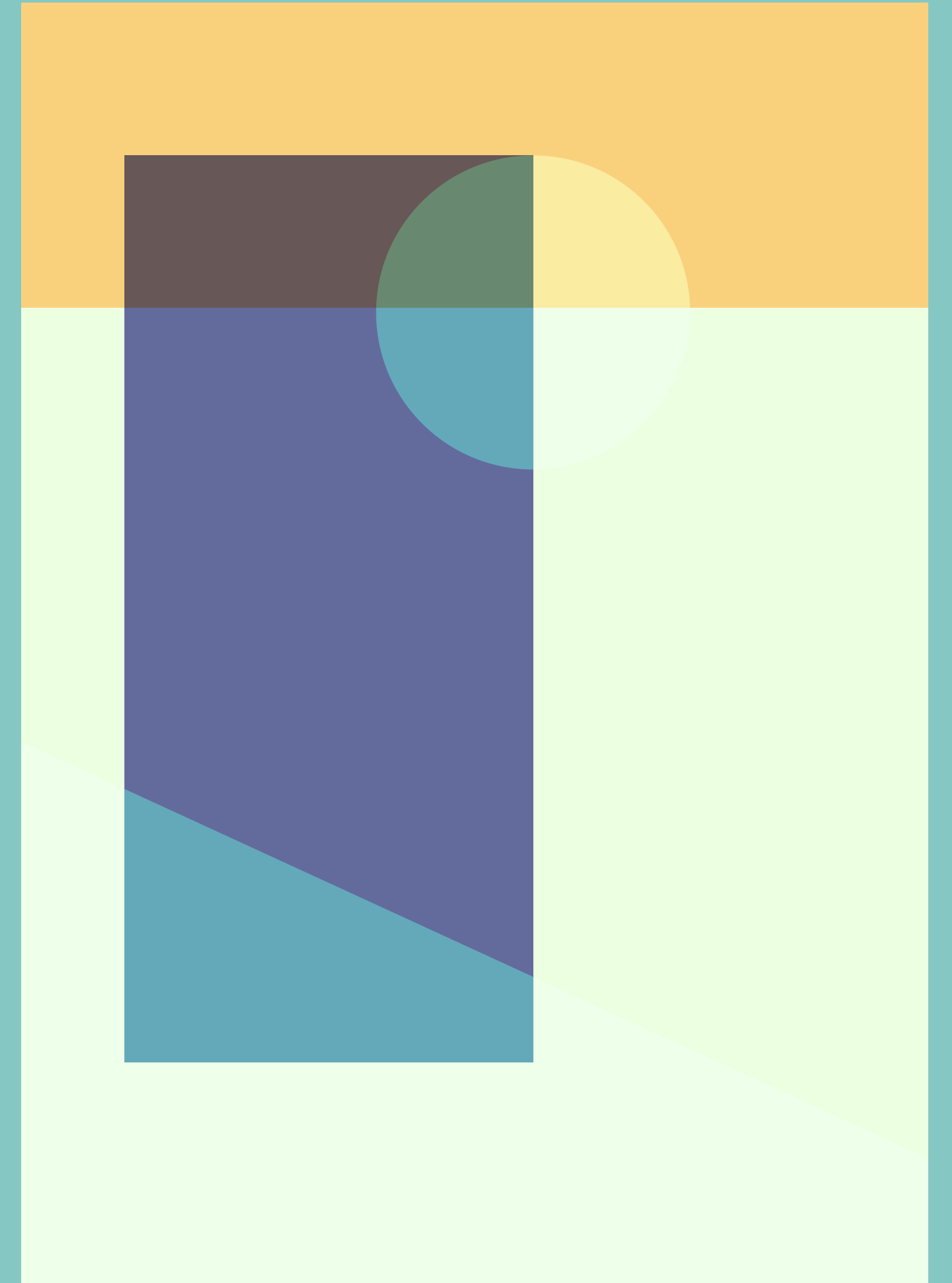
```
1  /// Verify an ECDSA Signature
2  fn verify_certificate(cert: &Certificate) -> Result<(), CryptoError> {
3      let params = CurveParams {
4          p: cert.public_key.curve.p,
5          n: cert.public_key.curve.n,
6          g: cert.public_key.curve.g,
7      };
8
9      ecdsa_verify(&cert.signature, &cert.public_key.point, &params)
10 }
11
```

11:1 Rust Z»



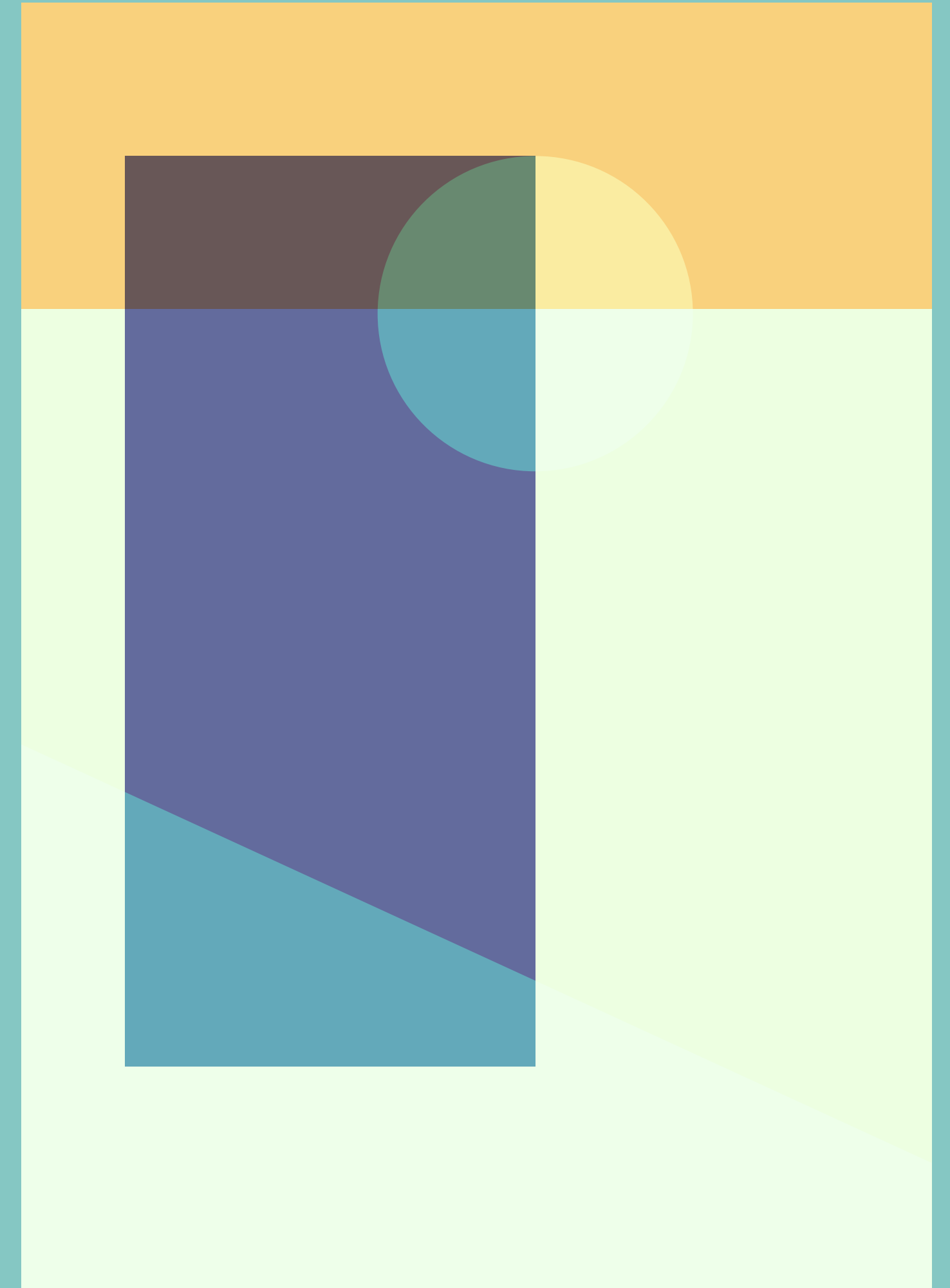
```
1  /// Verify an ECDSA Signature
2  fn verify_certificate(cert: &Certificate) -> Result<(), CryptoError> {
3      let params = CurveParams {
4          p: cert.public_key.curve.p,
5          n: cert.public_key.curve.n,
6          g: cert.public_key.curve.g,
7      };
8
9      ecdsa_verify(&cert.signature, &cert.public_key.point, &params)
10 }
11
```


The Differences



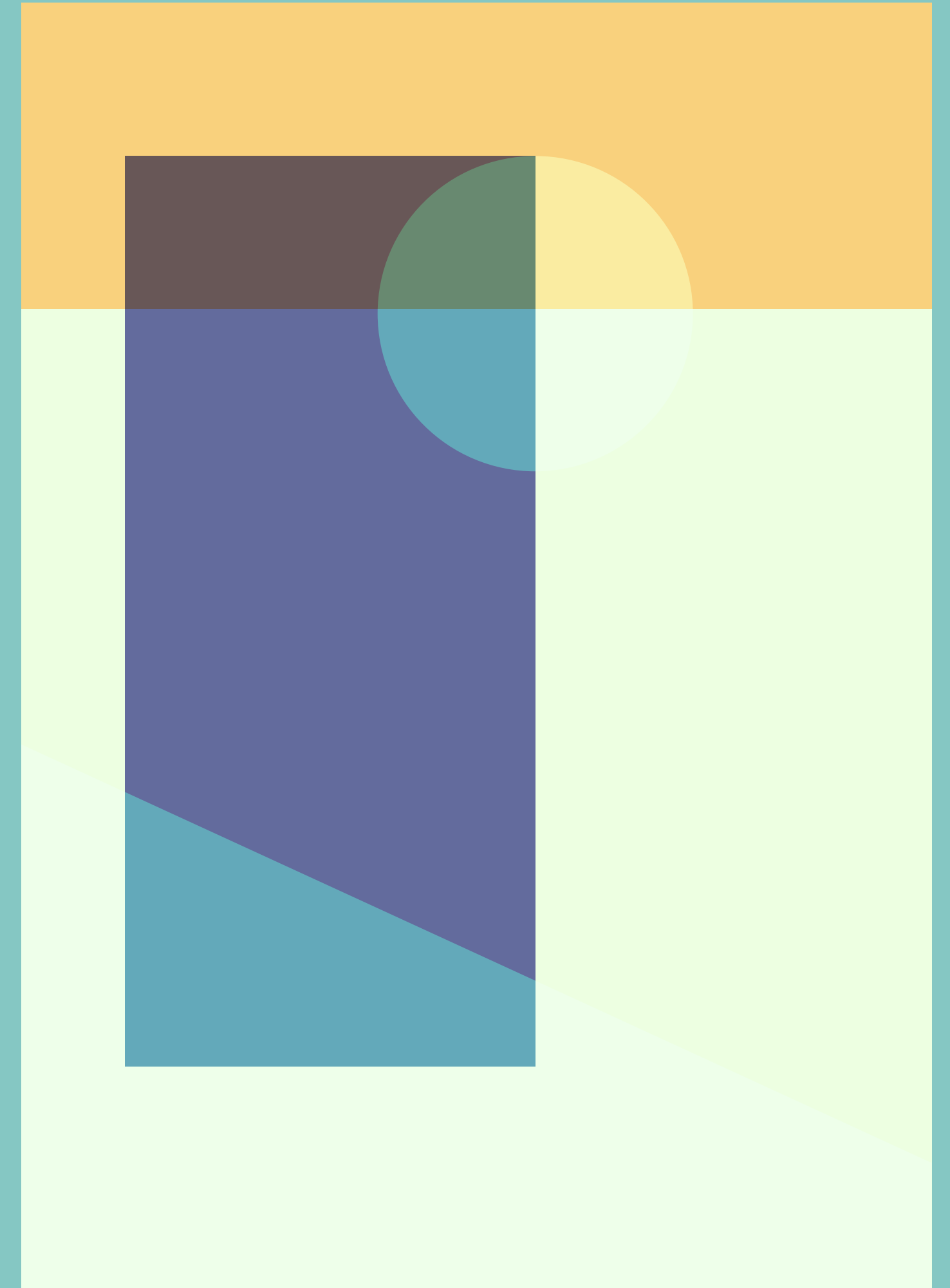
The Differences

- There is not always a bug to be found!



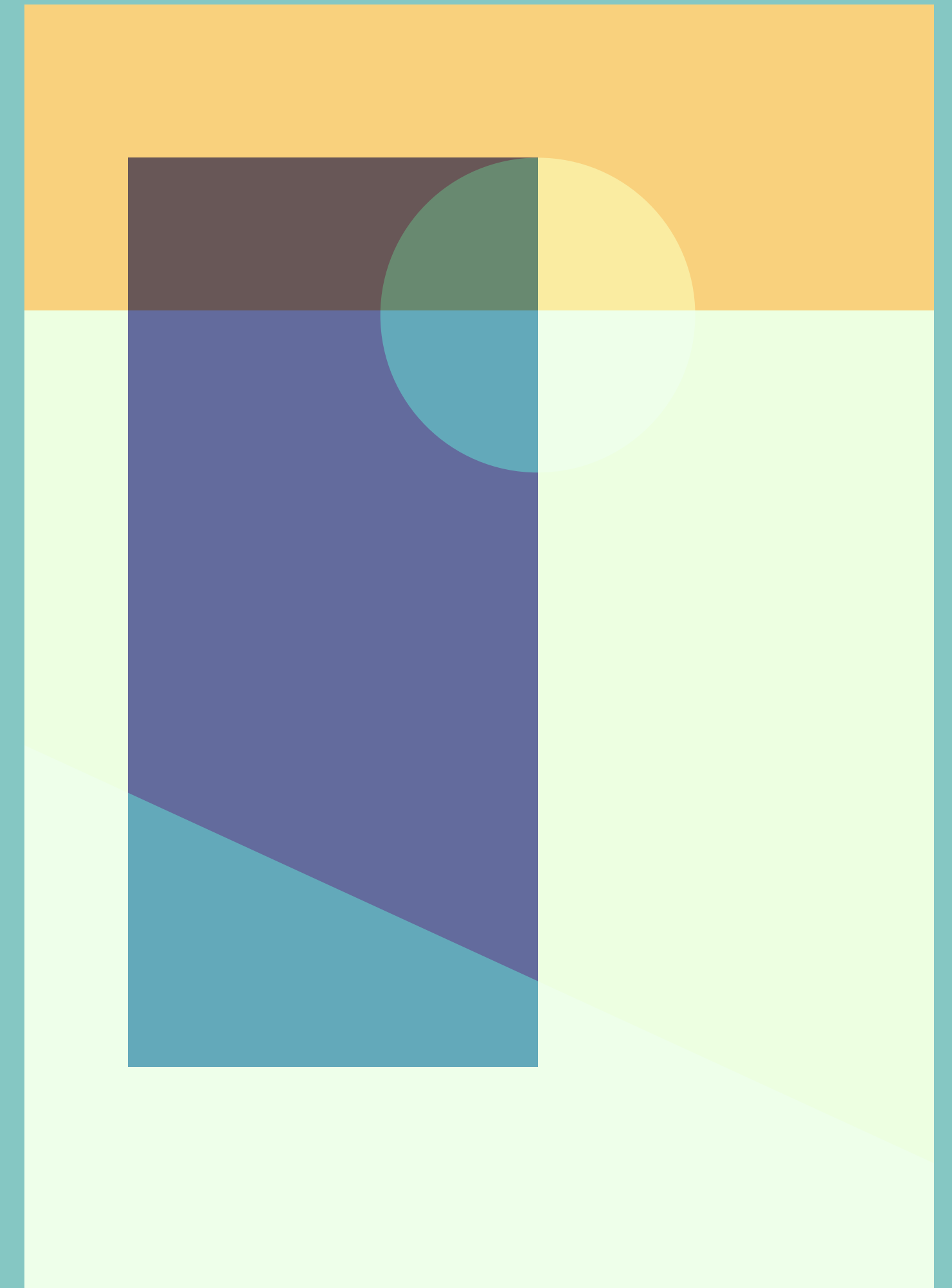
The Differences

- There is not always a bug to be found!
- We often do not require working exploits



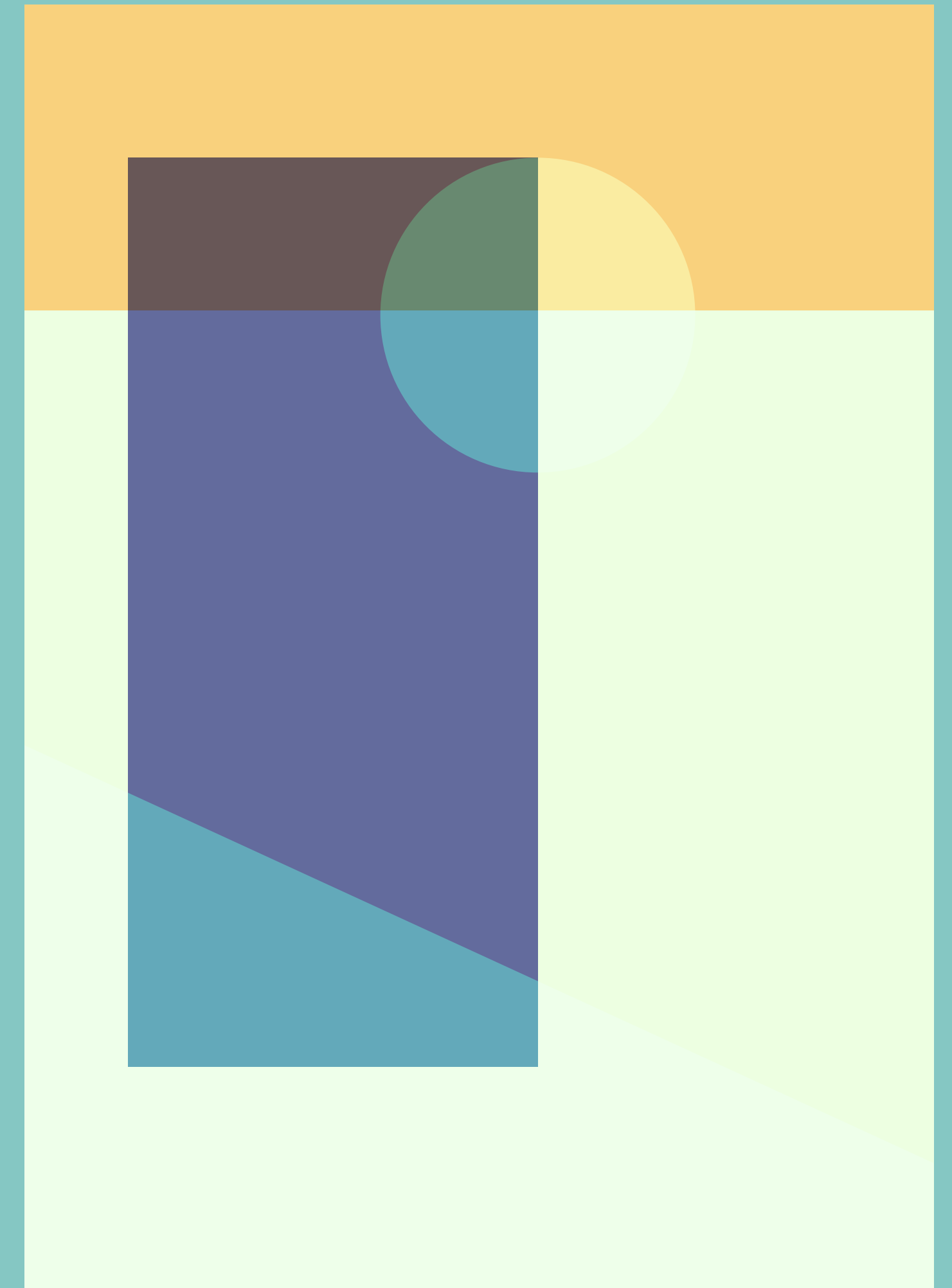
The Differences

- There is not always a bug to be found!
- We often do not require working exploits
- The value of our work is (partially) in non-technical discussions of the problem and proposed solution



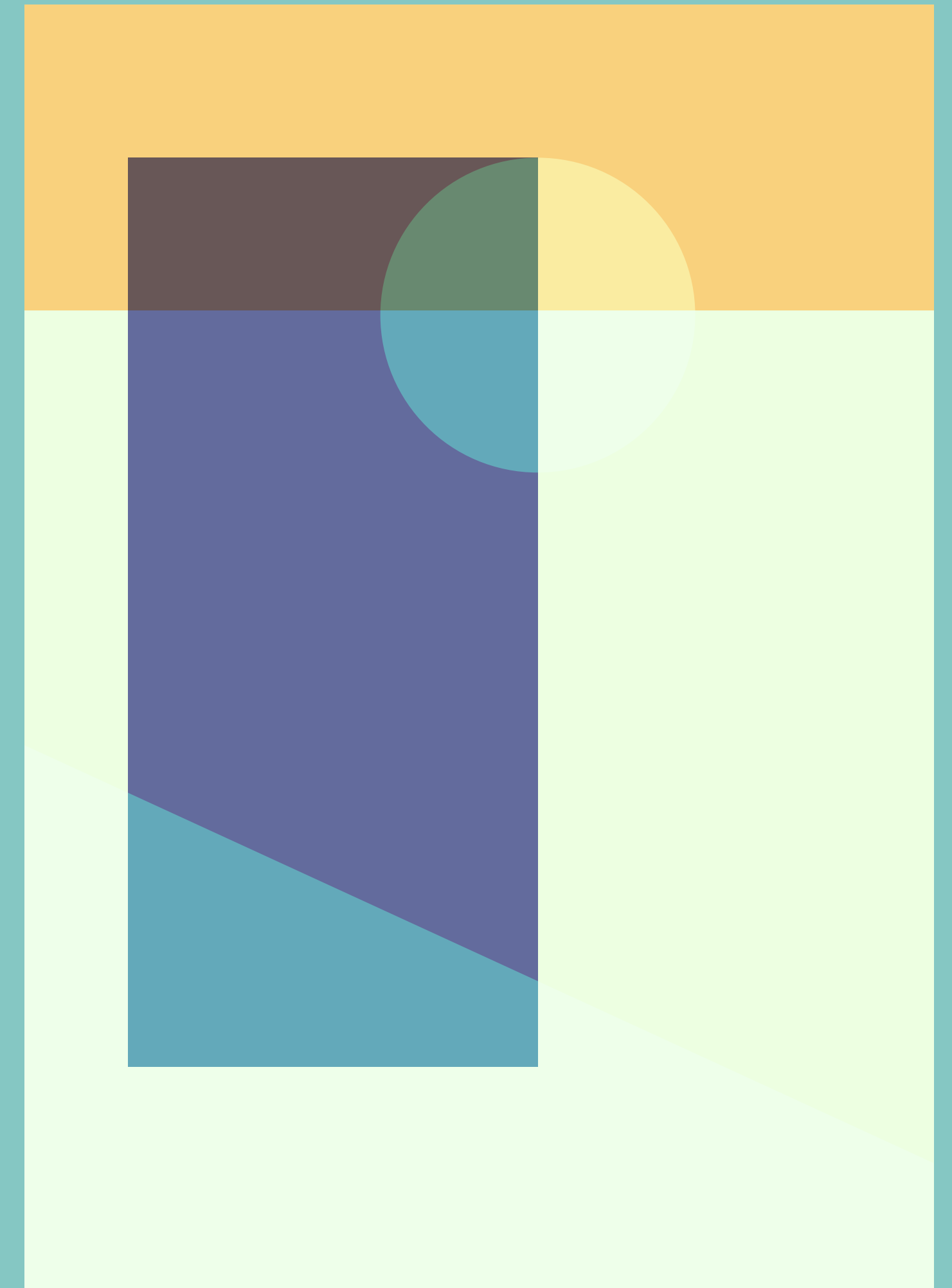
The Differences

- There is not always a bug to be found!
- We often do not require working exploits
- The value of our work is (partially) in non-technical discussions of the problem and proposed solution



The Differences

- There is not always a bug to be found!
- We often do not require working exploits
- The value of our work is (partially) in non-technical discussions of the problem and proposed solution
- I've yet to use **LLL** during my day job



Public Reports

Public Reports

- WhatsApp Message Summarization Service (Meta)
- Security Review of RSA Blind Signatures with Public Metadata (Google)
- Rust Cryptography Review (Entropy)

Public Reports

- WhatsApp Message Summarization Service (Meta)
- Security Review of RSA Blind Signatures with Public Metadata (Google)
- Rust Cryptography Review (Entropy)

2025

Google Private AI Compute Review
VetKeys Cryptography Review
Meta Whatsapp message summarization service
VeChain JavaScript SDK Cryptography and Security Review

2024

WhatsApp Contacts Security Assessment
Keyfork Implementation Review
Google Privacy Sandbox Aggregation Service and Coordinator
Aleo snarkOS Implementation and Consensus Mechanism Review

2023

Security Review of RSA Blind Signatures with Public Metadata
Aleo snarkVM Implementation Review
WhatsApp Auditable Key Directory (AKD) Implementation Review
Zcash FROST Security Assessment
Entropy/Rust Cryptography Review

Public Reports

- WhatsApp Message Summarization Service (Meta)
- Security Review of RSA Blind Signatures with Public Metadata (Google)
- Rust Cryptography Review (Entropy)

2025

Google Private AI Compute Review

VetKeys Cryptography Review

Meta Whatsapp message summarization service

VeChain JavaScript SDK Cryptography and Security Review

2024

WhatsApp Contacts Security Assessment

Keyfork Implementation Review

Google Privacy Sandbox Aggregation Service and Coordinator

Aleo snarkOS Implementation and Consensus Mechanism Review

2023

Security Review of RSA Blind Signatures with Public Metadata

Aleo snarkVM Implementation Review

WhatsApp Auditable Key Directory (AKD) Implementation Review

Zcash FROST Security Assessment

Entropy/Rust Cryptography Review

Thank

You

Questions?