

ZK Email: Zuthailand Workshop

**Aayush Gupta,
Sora Suegami**

with Aditya Bisht, Dimitri Dumonet,
Saul Garcia, John Guilding, Wataru
Shinohara, Shreyas Londhe, Shubham
Gupta, Prakhar Singh, Eesha Irfan

Roadmap

Basics

New Registry and SDK

Account Recovery

Devtooling

ZK Email: High Level Primitive

Emails are **signed** according to **DKIM protocol**

```
rsa_sign(sha256(from:<>, to:<>, subject:<>,
<body hash>, ...), RSA private key of domain)
```

ZK Email proves the full signature within a proof, along with parsing and selective disclosure!

ZK Email proofs are:

private

Keep arbitrary data private

provenant

Verify the data from the web2
service's mail server directly

portable

Move proofs onto chains

Whistleblowing

To: potus@gov.com
From: [REDACTED]@gov.com
Subject: Whistle Blowing

For example, you can prove you have an email from [REDACTED]@gov.com without disclosing your full address. We can confirm the email has mentions a name, place etc and confirm who we sent/recieved it from potus@gmail.com



Decentralized Domain Marketplace

zkp2p.xyz

Prove you own a Namecheap transfer
email, and build marketplace atop it



Log In

Popular Domains

Highest priced domains



encumbrance.net

4,444 ETH
0 bids

Bid Now



intelligenceage.xyz

32 ETH
2 bids

Bid Now



wethwhale.com

20 ETH
0 bids

Bid Now

Roadmap

Basics

New Registry and SDK

Account Recovery

Devtooling

zk email proof infrastructure

Registry

Reuse proofs that other people have already defined

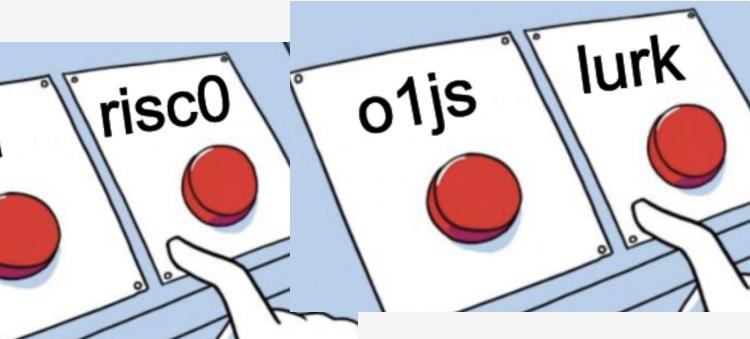
SDK

Easily use proofs from any proof system in your app

Apps

What are people building with this?

Don't make app devs choose



Registry

registry

List of community submitted ZK Email blueprints
that can be dropped into your project

Read Guide Create Blueprint

ZK

Blueprints

Proof Of Devcon Rejection Compiled

saugardev / devcon-rejection-proof

Verifies that an applicant received a rejection email for their Devcon proposal.

Extractable values: recipient_name | proposal_title | rejection_line | ...

Updated 22 hours ago

Search blueprints... Filters

Registry: Create New Patterns in 5 Minutes

Submit Blueprint

Create, compile and share blueprints easily by filling the following details

Pattern name

Proof of Devcon Rejection

Registry: Add example email to parse

Upload test .eml



[Click to upload](#) or drag and drop
(.eml format)

Our AI will autofill fields based on contents inside your mail. Don't worry you can edit them later

Registry: Auto-extract From Sample Email

Sender domain

twitter.com

Max email header length

1024

Must be a multiple of 64

Max email body length

4032

Must be a multiple of 64. If you have a Email Body Cutoff Value, it should be the length of the body after that value

Registry: AI to Automatically Define Regexes

AI auto extraction

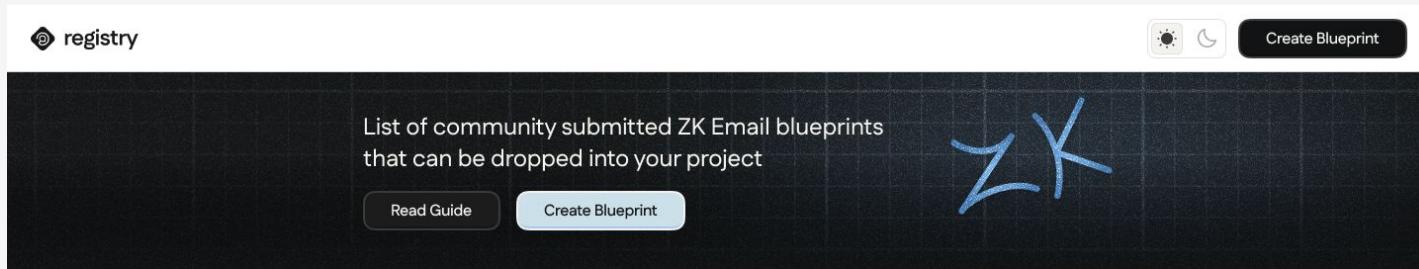
Use our AI to magically extract the fields you want

❖ Generate Fields

Registry: Final Config

```
{  
    "name": "proposal_title",  
    "parts": [  
        {  
            "is_public": false,  
            "regex_def": "your proposal \\""  
        },  
        {  
            "is_public": true,  
            "regex_def": "[^\"]*"  
        }  
    ],  
    "location": "body",  
    "maxLength": 192  
},  
{  
    "name": "rejection_line",  
    "parts": [  
        {  
            "is_public": false,  
            "regex_def": "we regret to inform you that "  
        },  
        {  
            "is_public": true,  
            "regex_def": "we were unable to accept this submission"  
        }  
    ],  
    "location": "body",  
    "maxLength": 64  
}  
,  
{"version": "v2",  
 "dkimSelector": "kh3h37a2jhlhfkth4f6smd3zlfz4sfffz",  
 "senderDomain": "devcon.org",  
 "enableMasking": false,  
 "externalInputs": [],  
 "emailBodyMaxLength": 5120,  
 "ignoreBodyHashCheck": false,  
 "shaPrecomputeSelector": "padding: 20px 28px 0 28px;"}
```

Registry: Easily View All Existing Proofs



Blueprints

Search blueprints...

Filters

Proof Of Devcon Rejection

✓ Compiled

972

Star 972

saugardev / devcon-rejection-proof



Verifies that an applicant received a rejection email for their Devcon proposal.

Extractable values: recipient_name | proposal_title | rejection_line | ...

Updated 22 hours ago

Proof Of Devcon Acceptance

✗ In Progress

1973

Star 973

saugardev / devcon-acceptance-proof



Since the domain g.ucla.edu doesn't sign emails, this will instead create a proof when someone receives an email from my dhruvpareek883@gmail.com email address. Submit an EML file which is an email received from dhruvpareek883@gmail.com with contents that follow the format: "Your Email Address Is: XXXX"

Extractable values: recipient_name

Updated 22 hours ago

Registry = “Vercel” of ZK Email: Autodeploy interface

The screenshot shows the Registry interface with the following details:

- Header:** A navigation bar with a logo, search, and a "Create Blueprint" button.
- Blueprint Overview:**
 - Name:** Proof Of Devcon Rejection
 - Status:** Compiled
 - Metrics:** 972 views, 972 stars
 - Description:** Verifies that an applicant received a rejection email for their Devcon proposal.
 - Version:** v1.4.4 (Updated 22h ago)
 - Actions:** View all versions
- Generate Proof Flow:** A three-step process:
 - Connect emails (highlighted with a green dot)
 - Select emails
 - View and verify
- Connect Emails Section:**
 - Text: Connect your Gmail or upload an .eml file
 - Note: Note - Your google API key is kept locally and never sent out to any of our servers.
 - Buttons: G Connect Gmail account (dark button), OR, Click to upload and drag (with a file icon).

Registry: Automatically Filter Emails Client-side

The screenshot shows the Registry interface. At the top, there's a navigation bar with a logo, a search icon, and a "Create Blueprint" button. Below the header, the blueprint title "Proof Of Devcon Rejection" is displayed, along with a "Compiled" status badge, a search icon with "972" results, and a star icon with "972" stars.

The main content area describes the blueprint: "Verifies that an applicant received a rejection email for their Devcon proposal." Below this, the version "v1.4.4" is shown, updated 22h ago, with a "Latest" badge, and a link to "View all versions".

A modal window titled "Generate Proof" is open. It has three steps: "Connect emails" (green dot), "Select emails" (white dot), and "View and verify" (white dot). The "Select emails" step is active, showing a "Select Emails" section. This section instructs users to "Choose the emails you want to create proofs for. You can select multiple emails." A note states: "Note - If you select to create the proofs remotely, your emails will be sent to our secured service for proof generation. Emails will be deleted once the proofs are generated." Below this, a table lists selected emails:

Select	Validity	Sent on	Subject	Generated Input
<input type="checkbox"/>	✓	9/22/2024 06:45:18 PM	[GitHub] A third party OAuth has been added to your workspace	View Input
<input type="checkbox"/>	✓	9/22/2024 06:45:18 PM	[GitHub] A third party OAuth has been added to your workspace	View Input
<input type="checkbox"/>	✓	9/22/2024 06:45:18 PM	[GitHub] A third party OAuth has been added to your workspace	View Input

Registry: Share finished proof

Proof Details

Share Proof

Job ID	cm2etv9jk0001oo56mt0edku1
Blueprint	Proof of GitHub (v1.4.4)
Outputs	{ "username": "PrakharSingh0908!" } { "username": "PrakharSingh0908!" }
Sent on	9/22/2024 09:45:45 PM
Date created	11/22/2024 09:45:45 PM
Time taken	120 seconds
Status	Completed

Generated proof

```
{  
  "pi_a": [  
    "716132586708770782518771465266552222431773306304909390227234954450110519094",  
    "6035485806245916393791579072676477633274149005565914247772978761887341423916",  
    "  
  ],  
  "pi_b": [  
    "  
  ]  
}
```

Registry = “Github” of ZK Email: Version History, Forks

The screenshot displays a user interface for managing version history, similar to GitHub. At the top, there is a back arrow labeled "← Proof of Devcon Rejection" and a title "Version History". In the top right corner, a button indicates "11 Versions".

The main content area shows three versions of a document:

- v 1.4.4** (Latest): Updated 22h ago. Includes a "Report" button.
- v 1.4.3** (Draft): Updated 1d ago. Includes a "Report" button.
- v 1.4.2**: Updated 2mo ago. Includes a "Report" button.

Each version card contains a "[version update description]" placeholder and standard "Download" and "Fork" buttons.



registry.zk.email

List of community submitted ZK Email blueprints that can be dropped into your project

[Read Guide](#)

A hand-drawn style logo for "ZK" in blue ink, featuring a stylized "Z" and "K" connected by a diagonal line.

Blueprints

Search blueprints..

Filter and Sort

Proof of Devcon Acceptance ✓ Compiled

saugardev/DevconAcceptance [🔗](#)

Prove you got a Devcon acceptance email.

Extractable values: [subject_congrats](#)

Updated 1 days ago

Proof of Stripe payment ✓ Compiled

saugardev/ProofStripePayment [🔗](#)

Proof of Stripe payment

Extractable values: [merchant](#) [order_id](#)

Updated 1 days ago

SDK: Decides the proof system for you

```
import { createProver, parseEmail } from 'zk-email-sdk';

const blueprint = await sdk.getBlueprintById("saugardev/devcon-rejection");
const prover = blueprint.createProver({isLocal: false});
```



Proof of Pad Thai

[receipt](#)[food](#)[grab](#)[thailand](#)[pad](#)[thai](#)

johnguilding/proof-of-pad-thai [🔗](#)

Last modified: 17/10/2024

Prove you're an enthusiast for Thailand's national dish by submitting a proof of Grab receipt which contains reference to a Pad Thai order



Proof of Devcon Rejection

[devcon](#)[rejection](#)[talk](#)

saugardev/devcon-rejection-proof [🔗](#)

Last modified: 18/10/2024

Verifies that an applicant received a rejection email for their Devcon proposal.

Extractable values: recipient_name, proposal_title, rejection_line

[Try it out](#)[Download Example Project](#)[Download .zkey](#)[View Parameters](#)

Roadmap

Proofs of Received Emails

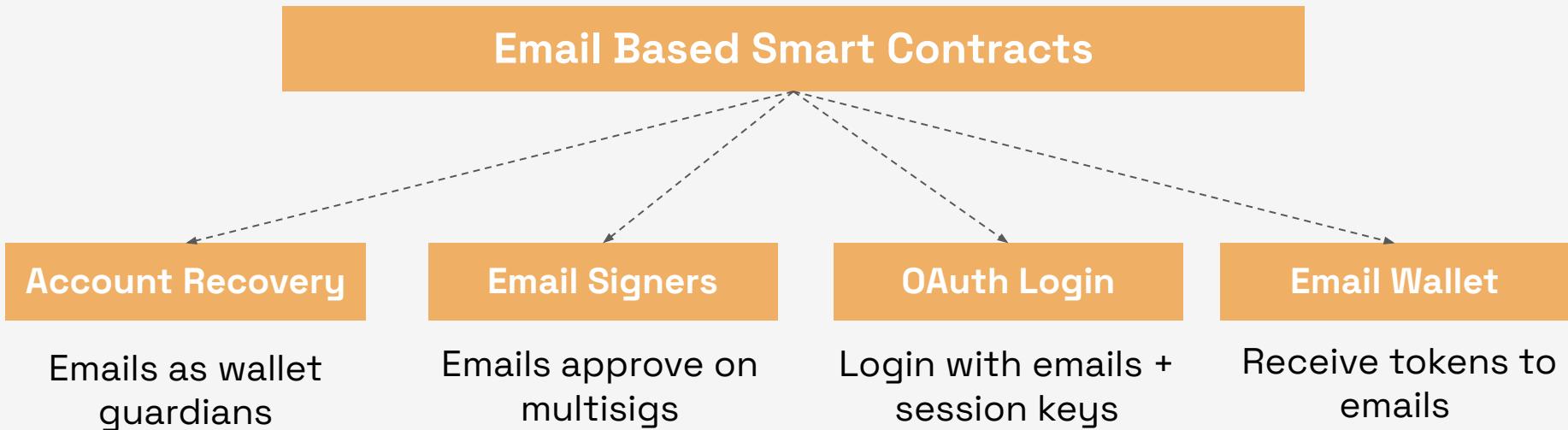
New Registry and SDK

Account Recovery

Devtooling

Email Triggered Transactions

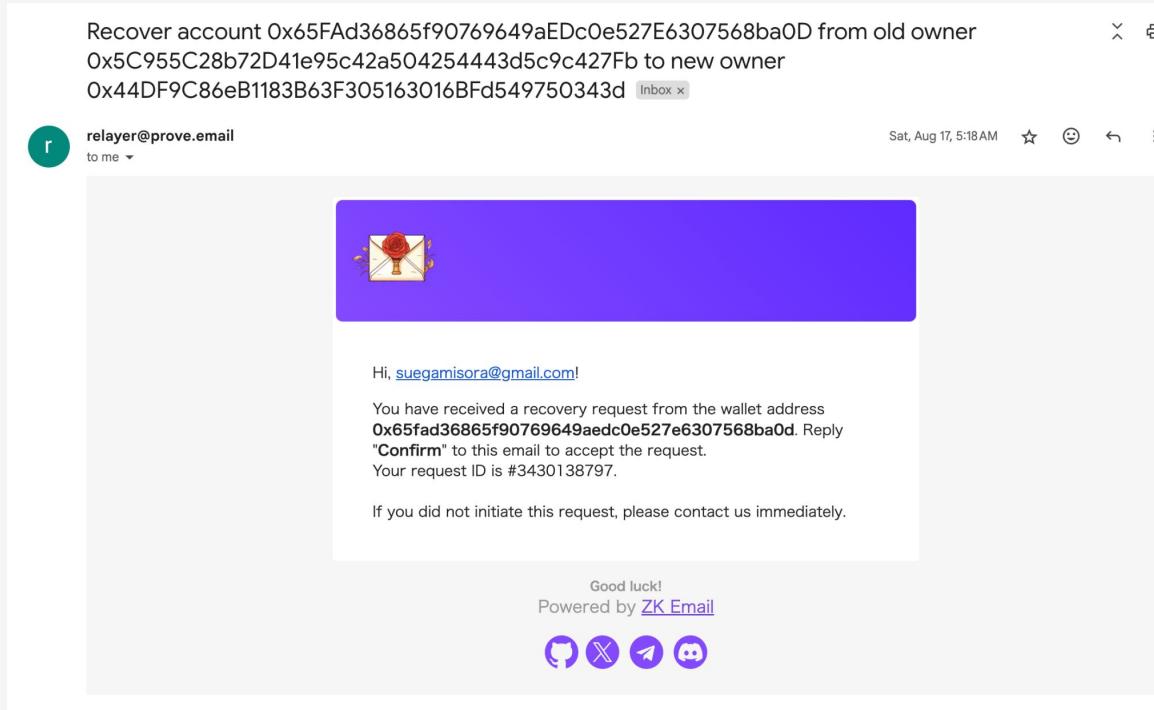
ZK Email improves security & UX of smart accounts



Send Commands to Smart Contracts through Emails

Users authorize transactions by simply replying to emails.

Recover account 0x65FAd36865f90769649aEdc0e527E6307568ba0D from old owner
0x5C955C28b72D41e95c42a504254443d5c9c427Fb to new owner
0x44DF9C86eB1183B63F305163016BFd549750343d [Inbox](#)



relayer@prove.email
to me ▾

Sat, Aug 17, 5:18 AM

Hi, suegamisora@gmail.com!

You have received a recovery request from the wallet address **0x65fad36865f90769649aedc0e527e6307568ba0d**. Reply "Confirm" to this email to accept the request.
Your request ID is #3430138797.

If you did not initiate this request, please contact us immediately.

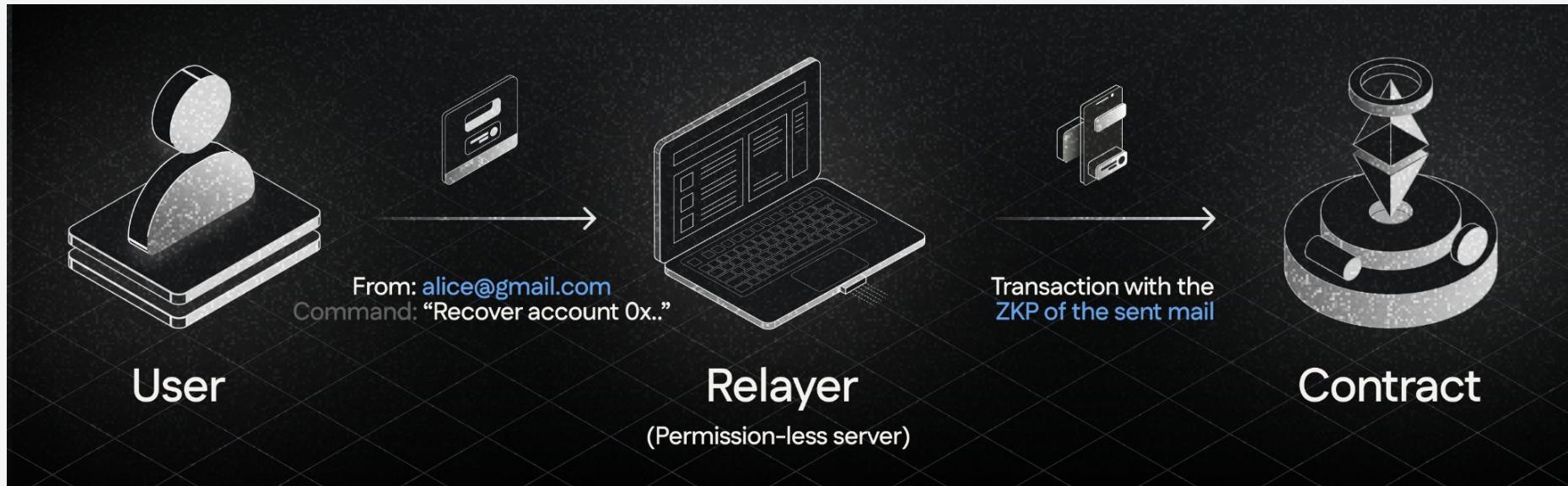
Good luck!
Powered by [ZK Email](#)

 Sora Suegami <suegamisora@gmail.com>
to relayer ▾

Sat, Aug 17, 5:18 AM

Confirm.

Send Commands to Smart Contracts through Emails



Send Commands to Smart Contracts through Emails

Account Code

Accept guardian request for 0x6EB0660a96a01Cef20282d194656c8457d562F98 Code
2d6837cb9ef9d2b2569e6aac61d3e0214d7f5205225750bc4a0eae140fb3b4f7 Inbox x

 relayer@prove.email
to me ▾

Sun, Sep 1, 2:43 PM ☆ 😊 ↶ ⋮



Hi, [suegamisora@gmail.com!](mailto:suegamisora@gmail.com)

You have received a guardian request from the wallet address **0x6eb0660a96a01cef20282d194656c8457d562f98**. Reply "Confirm" to this email to accept the request.
Your request ID is #2583762894.

If you did not initiate this request, please contact us immediately.

Account Code Check -> Privacy & Decentralization

- **hash(email address, code)** \Rightarrow eth address: **email privacy on-chain**
- prove **availability** to user in email \Rightarrow **access cannot be withheld**
- **relayer decentralization:** fault tolerant and censorship resistant

Email Replies or JWTs



Hi,

You have received a guardian acceptance request ...

Reply "**Confirm**" to this email to accept the request. Your request ID is d116b3ba-dd09-45b2-9051-d212e581308b.

If you did not initiate this request, please contact us immediately.

Cheers,

The ZK Email Team

Powered by [ZK Email](#)



JWT-Wallet

Welcome to JWT-Wallet! Follow these steps to get started:

1. [Enter a command](#) in the input field below (e.g., "Send 0.12 ETH to 0x1234...")
2. The [Google Sign-In button](#) will become active once you've entered a command
3. [Click the Google Sign-In button](#) to authenticate and generate a JWT
4. [Check the console](#) for the decoded JWT information

Enter your command here

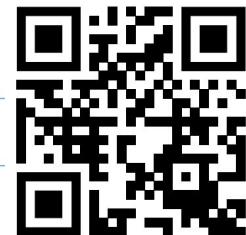
[Sign in with Google](#)

JWT
Generation

> Proof
Generation

> Proof
Complete

> Submit to
Contract



Email account recovery

Use email guardians to recover your account

[Blog](#)[Docs](#)[Demos](#)[Contact](#)

Email Recovery Demo

Assigned Guardians must reply back to an email to enable wallet recovery to a new address.

[Set Up](#)[Recover](#)

Gnosis Safe

Copy the link and import into your Safe wallet

[Safe Wallet Flow](#)

Test Wallet

Connect to see the test wallet flow

[Burner Safe Flow \(v1.4.1\)](#)

Email Recovery: Social Recovery using Emails

Just like a bank account or paypal,
anyone with an email address can help
recover your account.

Email Recovery with Passkeys

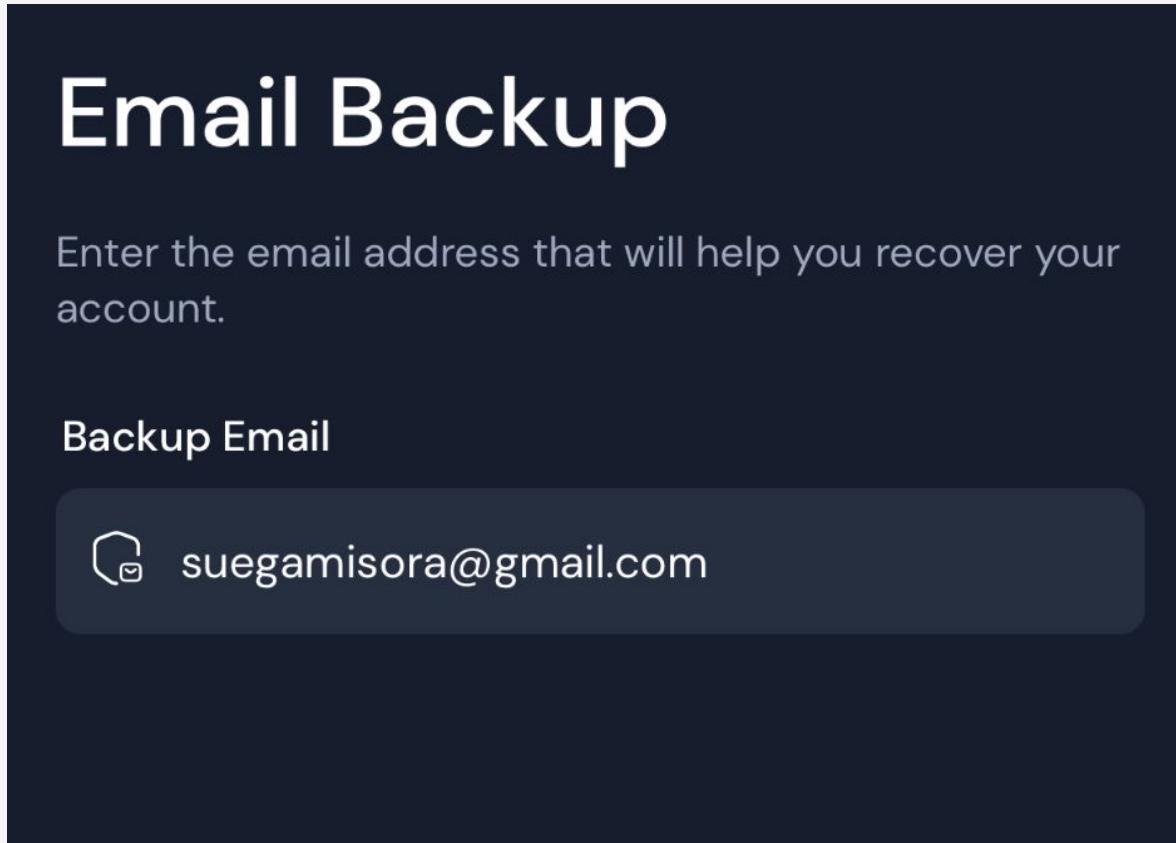


1. Configure recovery settings

Configure necessary recovery settings such as guardians, threshold, recovery timelock etc

How it works

1. Configure recovery settings



Clave

1. Configure recovery settings

Configure necessary recovery settings such as guardians, threshold, recovery timelock etc

2. Accept guardian

Guardians explicitly accept being a guardian.
Prevents broken recovery setups

How it works

2. Accept Guardian

[Reply Needed] Recovery: Acceptance Request Inbox x ⋮ 🖨️

 **bosisalim8@gmail.com** to me ▼ 12:55 PM (7 hours ago) ☆ 😊 ↶ ⋮

 zkemail

Hi, suegamisora@gmail.com!

You have received an guardian request from the wallet address **0xf51861615dbc0661f3bafedbe878d486117bb217**. Reply "**Confirm**" to this email to accept the request. Your request ID is #159567672.

If you did not initiate this request, please contact us immediately.

Cheers,
The ZK Email Team

Powered by [ZK Email](#)

𝕏 𝕏 𝕏 𝕏

2. Accept Guardian



Sora Suegami

Hi, suegamisora@gmail.com! You have received a guardian request from the wallet address 0xf51861615dbc0661f3bafedbe878d486117bb217. Reply "Confirm" to this email

12:58 PM (7 hours ago)



bosisalim8@gmail.com

Hi, suegamisora@gmail.com! We have received your following request: Accept guardian request for 0xF51861615dBC0661f3BAfedBE878d486117bb217 Code 02540b2dbf5f2475

12:58 PM (7 hours ago)



bosisalim8@gmail.com

to me ▾

12:59 PM (7 hours ago)



Hi, [suegamisora@gmail.com!](mailto:suegamisora@gmail.com)

Your guardian request for the wallet address
0xf51861615dbc0661f3bafedbe878d486117bb217. Your request ID is #**159567672** is now complete.

Cheers,
The ZK Email Team

1. Configure recovery settings

Configure necessary recovery settings such as guardians, threshold, recovery timelock etc

2. Accept guardian

Guardians explicitly accept being a guardian.
Prevents broken recovery setups

3. Process recovery

Guardians approve a recovery request. Can complete recovery once threshold is met

How it works

2.5. You lose your key



3. Process Recovery

Recover your account

Type your guardian email address to start recovery process

Guardian Email Address



suegamisora@gmail.com

3. Process Recovery

[Reply Needed] Recovery: Recovery Request Inbox x ⋮ 🖨️

 **bosisalim8@gmail.com** to me ⋮

1:01PM (7 hours ago) ☆ 😊 ↶ ⋮



Hi, suegamisora@gmail.com!

You have received a recovery request from the wallet address **0xf51861615dbc0661f3bafedbe878d486117bb217**. Reply "**Confirm**" to this email to accept the request. Your request ID is #1335835231.

If you did not initiate this request, please contact us immediately.

Cheers,
The ZK Email Team

Powered by [ZK Email](#)

3. Process Recovery



Sora Suegami

Hi, suegamisora@gmail.com! You have received a recovery request from the wallet address 0xf51861615dbc0661f3bafedbe878d486117bb217. Reply "Confirm" to this email

1:02 PM (7 hours ago)



bosisalim8@gmail.com

Hi, suegamisora@gmail.com! We have received your following request: Recover account 0xF51861615dBC0661f3BAfedBE878d486117bb217 using recovery hash 0x83f4490f8...

1:02PM (7 hours ago)



bosisalim8@gmail.com

to me ▾

1:03 PM (7 hours ago)



Hi, suegamisora@gmail.com!

Your recovery request for the wallet address
0xf51861615dbc0661f3bafedbe878d486117bb217 is successful. Your request ID is
#1335835231 is now complete.

Cheers,
The ZK Email Team

1. Configure recovery settings

Configure necessary recovery settings such as guardians, threshold, recovery timelock etc

2. Accept guardian

Guardians explicitly accept being a guardian.
Prevents broken recovery setups

3. Process recovery

Guardians approve a recovery request. Can complete recovery once threshold is met

4. Complete recovery

Separate step to allow for recovery delays. Protects against malicious recovery attempts

How it works

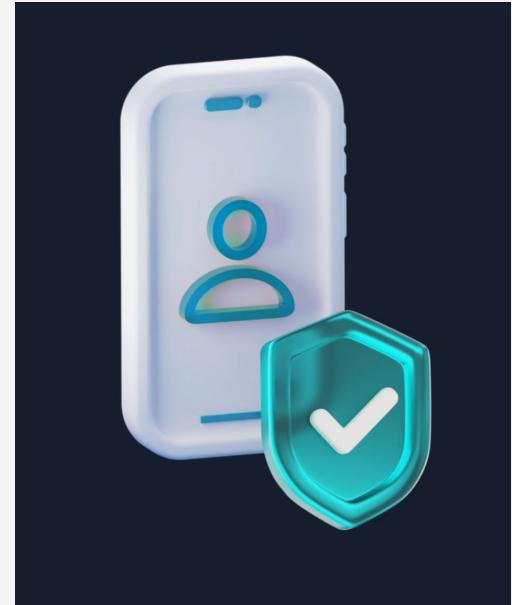
2.5. You lose your key



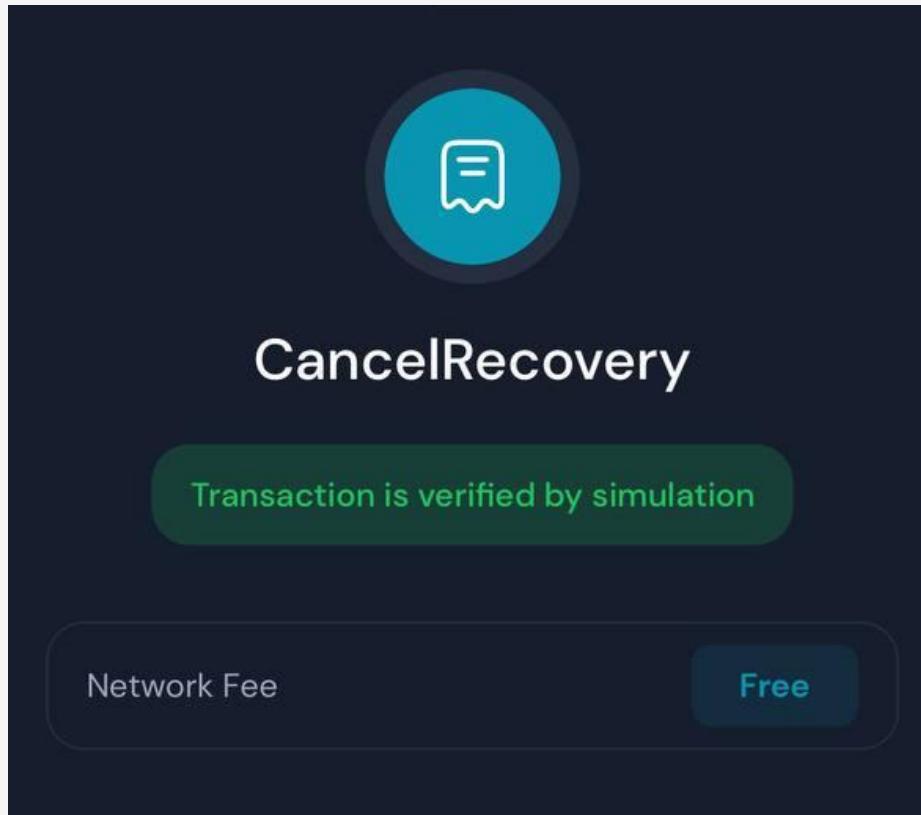
4. Complete Recovery



N minutes later...



Cancel recovery if the guardian is hacked



Roadmap

Basics

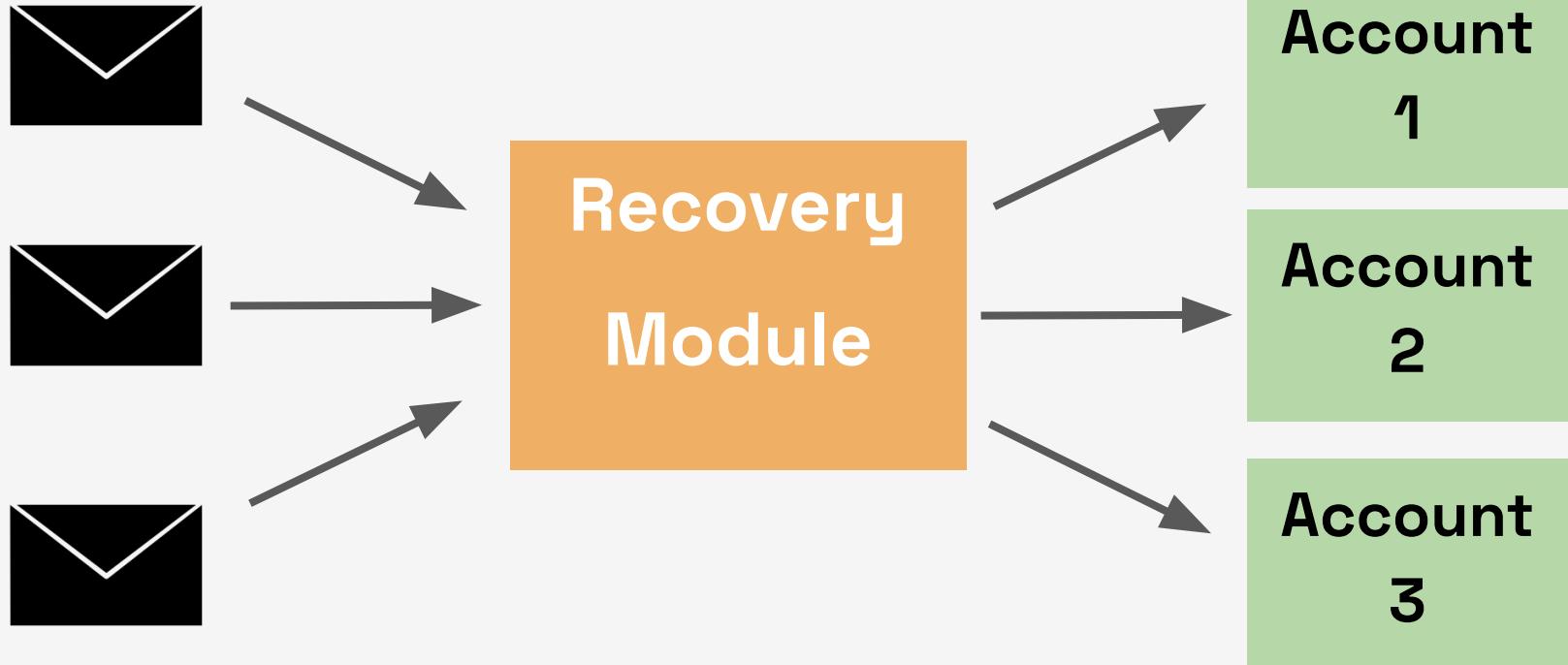
New Registry and SDK

Account Recovery

Devtooling

Email Recovery as ERC7579 Module

Recover existing Safes or **any smart account** via ERC7579 standard



Email Recovery as ERC7579 Module

Recover existing Safes or any smart account via ERC7579 standard

1. Configure recovery settings



Install the module on each account.

2. Accept guardian



Call

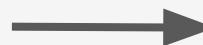
3. Process recovery



Call

`${relayerApiUrl}/recoveryRequest`

4. Complete recovery



Call

`${relayerApiUrl}/completeRequest`

Add account recovery to any smart account

```
const userOpHash = await smartAccountClient.installModule({  
  type: "executor",  
  address: universalEmailRecoveryModuleAddress,  
  context: moduleData,  
});  
  
const receipt = await pimlicoClient.waitForUserOperationReceipt({  
  hash: userOpHash,  
});
```



docs.zk.email

Email Tx Builder: Generic Solidity Proofs + Relayer

```
function recoverySubjectTemplates() public pure returns
(string[][] memory) {
    string[][] memory templates = new string[][](1);
    templates[0] = new string[](11);
    templates[0][0] = "Recover";
    templates[0][1] = "account";
    templates[0][2] = "{ethAddr}";
    templates[0][3] = "to";
    templates[0][4] = "new";
    templates[0][5] = "owner";
    templates[0][6] = "{ethAddr}";
    templates[0][7] = "using";
    templates[0][8] = "recovery";
    templates[0][9] = "module";
    templates[0][10] = "{ethAddr}";

    return templates;
}
```

- Example API call for `/submit`

2

```
{
    "contractAddress": "0xec3a1eAD94BDc7527Aa807B97Ff7E3A2cBCbC75f",
    "dkimContractAddress": "0x0a3921e8Bb2d682f92a39cB006b77AEc8939d1B9",
    "accountCode": "0x22a2d51a892f866cf3c6cc4e138ba87a8a5059a1d80dea5b8ee8232034a105",
    "codeExistsInEmail": true,
    "functionAbi": abi,
    "commandTemplate": "Emit string {string}",
    "commandParams": ["testing"],
    "templateId": "0x25d6c3eada7b2926c822bbfebfc3173123afb205cf093a8cae6622a56712f8a",
    "remainingArgs": [
        { "Address": "0x9401296121FC9B78F84fc856B1F8dC88f4415B2e" },
        { "Uint": "0x0" }
    ],
    "emailAddress": "bisht.s.aditya@gmail.com",
    "subject": "Testing",
    "body": "Testing",
    "chain": "baseSepolia"
}
```

Email Transaction Builder



docs.zk.email/ > Email Tx Auth > Quickstart

Built with Email Transaction Builder



Account Recovery

Emails as wallet
guardians



Email Signers

Emails approve on
multisigs



OAuth Login

Login with emails +
session keys



Email Wallet

Receive tokens to
emails

Roadmap

Basics

New Registry and SDK

Email Triggered Transactions

Devtooling

Booths & Events

Museum Exhibit at Cursive

Nov 12-15

Cursive Connections Museum, Main Venue (SKNCC)

Interpretive exhibit representing ZK Email!

ZK Email Sessions at PSE Impact Booth

Nov 12, 14:00-17:00 and Nov 13, 11:00-14:00

Main Venue (SKNCC)

Come meet the team, solve challenges, and try out our new tech with the folks who built it!

ZK Email Impact Booth

Nov 14-15 all day

Main Venue (SKNCC)

Come meet the team, solve challenges, and try out our new tech with the folks who built it!

Discussion at AA Community Hub

Nov 15, 14:15-15:00

Main Venue (SKNCC)

Demo and discussion on ZK Email for Account Abstraction.

Talks & Panels

Mainstage Talk on ZK Email

Nov 13, 10:30 AM

Main Venue (SKNCC), Stage 3

All the ZK Email updates — community progress, fresh devtools, fast proofs, and new mainnet launches!

SDKs & DevTooling Workshop

Nov 14, 13:30-15:00

Main Venue (SKNCC), Classroom E

Build your own email proofs, email triggered transactions, and/or account recovery — Get a headstart on your hackathon projects!

Verifiable Data: No BS Debate

Nov 14, 4:15 PM - 4:45 PM

VLayer Panel, JW Marriott Hotel Bangkok

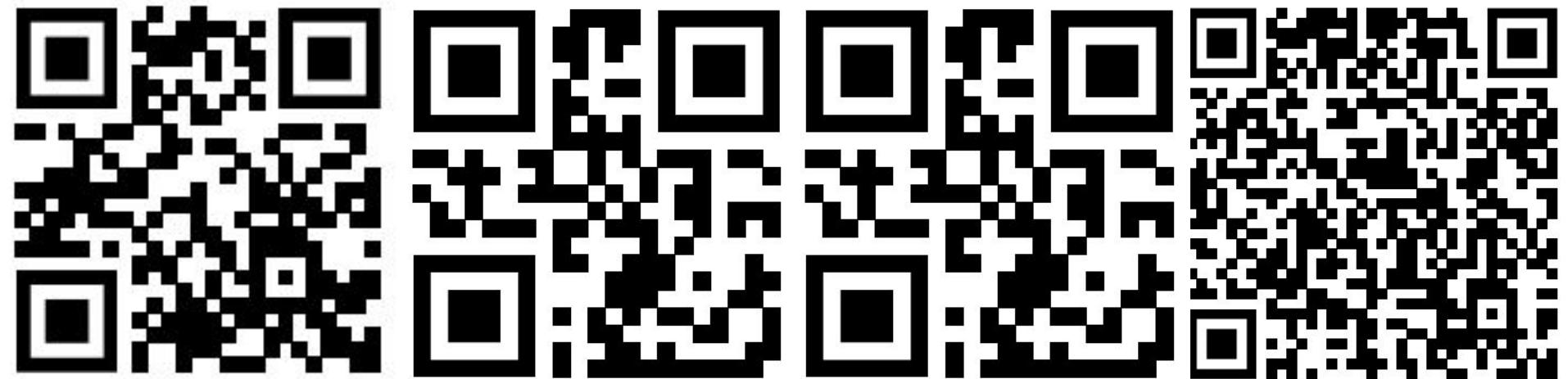
A debate on verifiable data.

ProgCrypto Lightning Talk

Nov 15, 12:30 PM - 2:00 PM

Progcrypto community led session, Main Venue (SKNCC), Breakout 2

A 7-9 minute lightning talk with ZK Email updates for ProgCrypto!



t.me/zkemail

https://zk.email

docs.zk.email/

Slides