Survey for Cryptographic Practices in Ethereum Smart Contracts

We would like to hear about your experience in implementing cryptographic tasks in Ethereum smart contracts. **Except for the first four questions, all other questions are optional. If there are any questions you don't understand, please skip them or choose "I don't understand this question".**

* Indicates required question

1. Are you a smart contract practitioner? *

- ◯ Yes
- ◯ No

2. What is your main role as a smart contract practitioner? *

- ◯ Development
- ◯ Testing
- ◯ Project management
- ◯ Research
- ◯ Other: _____

3. How many years of experience do you have in Solidity smart contract development/testing/project management/research/others? *

Your answer

4. What country do you currently reside in? *

Your answer

5. Which of the following Ethereum crypto APIs have you used? (Multiple Choice). In this survey, "Ethereum crypto APIs" refers to the crypto-related opcodes/precompiled contracts supported by Ethereum.

- [ ] KECCAK256 / SHA3 (https://www.evm.codes/#20)
- [ ] ecRecover (https://www.evm.codes/precompiled#0x01)
- [ ] SHA256 (https://www.evm.codes/precompiled#0x02)
- [ ] RIPEMD160 (https://www.evm.codes/precompiled#0x03)
- [ ] ModExp (https://www.evm.codes/precompiled#0x05)
- [ ] ECADD (https://www.evm.codes/precompiled#0x06)
- [ ] ECMUL (https://www.evm.codes/precompiled#0x07)
- [ ] ECPairing (https://www.evm.codes/precompiled#0x08)
- [ ] BLAKE2 (https://www.evm.codes/precompiled#0x09)
- [ ] None of them.
- [ ] I don't understand this question.
- [ ] Other:

6. Which of the following cryptographic primitives in smart contracts are you familiar with?

- [ ] Hash (including KECCAK256/SHA3, SHA2-256, RIPEMD-160, etc)
- [ ] Signature (including ECDSA, etc.)
- [ ] Zero-knowledge proof (including Plonk, etc.)
- [ ] I don't understand this question.
- [ ] Other: _____

7. Do you believe that implementing cryptographic tasks (such as signature verification) is more challenging than other common programming tasks in smart contracts?

- ( ) Yes (Go to Question 8)
- ( ) They are basically the same, but lie in different aspects (Go to Question 8)
- ( ) No (Go to Question 9)
- ( ) I don't understand this question. (Go to Question 9)
- ( ) Other: _____

8. Why do you think implementing cryptographic tasks is more challenging, or different from other common tasks in smart contracts?

Your answer

9. Have you encountered the following obstacles while accomplishing cryptographic tasks?

- [ ] Knowledge obstacles. For example, I don't understand the basic concepts and principles in cryptography/blockchain fields.
- [ ] Roadmap identification obstacles. For example, I struggle with identifying the appropriate template/crypto API to use.
- [ ] Template usage obstacles. For example, I encounter issues when trying to reuse the OpenZepplin's Library.
- [ ] Ethereum crypto API usage obstacles. For example, I encounter issues when trying to directly call the Ethereum precompiled contracts such as ecRecover.
- [ ] Security obstacle. For example, I am uncertain whether my implementation is secure.
- [ ] None of them.
- [ ] Other: _____

10. How do you obtain required knowledge for cryptographic tasks in smart contracts? (Multiple Choice)

- [ ] Official documentations, such as Solidity documentations, Ethereum Yellow Paper
- [ ] Documentations of the template provider, e.g., OpenZeppllin
- [ ] Online tutorials/blogs
- [ ] Q&A sites, such as Stack Overflow
- [ ] Technical books / papers
- [ ] I don't understand this question
- [ ] Other: _____

11. For cryptographic tasks that have available templates (for example, digital signature have been implemented in OpenZeppelin ECDSA and many other templates), do you prefer to use existing templates or implement them from scratch?

- [ ] I usually use existing templates.
- [ ] I usually implement cryptographic tasks from scratch.
- [ ] It depends on the complexity of the task. For complex tasks, I use existing templates.
- [ ] It depends on the complexity of the task. For simple tasks, I use existing templates.
- [ ] I don't understand this question.
- [ ] Other: _____

12. Are there any cryptographic tasks that you want to accomplish but currently lack available templates? If yes, please list them below. Otherwise, please skip this question.

Your answer

13. Do you think current Ethereum crypto APIs are good enough to support all your cryptographic tasks, in terms of functionality, flexibility, and usability? Please rate them on a scale of 1-5.
(1) Functionality： Do the APIs cover all functionalities I needed?
(2) Usability: Are the APIs easy to understand & use, with easily accessable documentation?

	1 (Very bad)	2 (Bad)	3 (Neutrality)	4 (Good)	5 (Very Good)
Functionality	○	○	○	○	○
Usability	○	○	○	○	○

14. Are there any cryptographic tasks that you wish to accomplish, but are currently unsupported by Ethereum's crypto APIs? If yes, please list them below. Otherwise, please skip this question.

Your answer

15. Do you think ensuring the security of cryptographic implementations is more challenging than securing other parts of a smart contract?

- ◯ Yes (Go to Question 16)
- ◯ They are basically the same, but lie in different aspects (Go to Question 16)
- ◯ No (Go to Question 17)
- ◯ I don't understand this question (Go to Question 17)
- ◯ Other:

16. Why do you think securing cryptographic Implementations is more challenging, or different from other common programming tasks?

Your answer

17. Which of the following crypto-related vulnerabilities in the Smart Contract Weakness Classification (SWC) list are you familiar with? (Multiple Choice)

- [ ] SWC-117: Signature Malleability (https://swcregistry.io/docs/SWC-117)
- [ ] SWC-120: Weak Sources of Randomness from Chain Attributes (https://swcregistry.io/docs/SWC-120)
- [ ] SWC-121: Missing Protection against Signature Replay Attacks (https://swcregistry.io/docs/SWC-121)
- [ ] SWC-122: Lack of Proper Signature Verification (https://swcregistry.io/docs/SWC-122)
- [ ] SWC-133: Hash Collisions With Multiple Variable Length Arguments (https://swcregistry.io/docs/SWC-133)
- [ ] None of them
- [ ] I don't understand this question
- [ ] Other: _____

18. Do you think the following types of tools / resources are good enough to support secure and efficient implementations of cryptographic tasks? Please rate them on a scale of 1 to 5.

	1 (Very bad)	2 (Bad)	3 (Neutrality)	4 (Good)	5 (Very Good)
Solidity and Ethereum documentation	○	○	○	○	○
Audited templates, e.g., Openzepplin's templates	○	○	○	○	○
Testing tools	○	○	○	○	○
Security audit tools	○	○	○	○	○

19. In addition to the four types of tools/resources mentioned in the previous question, are there any other types of support you desire? If yes, please list them below. Otherwise, please skip this question.

Your answer

20. If you have any final comments or questions for us, please kindly list them below.

Your answer

As an appreciation of your time and valuable inputs, we will give out $50 USDT Tokens to two randomly selected participants. If you want to enter the raffle, please kindly enter your address on Ethereum mainnet.

Your answer

Get link

Never submit passwords through Google Forms.