DRAM stablecoin

Date	August 2023
Auditors	Chingiz Mardanov

1 Executive Summary

This report presents the results of our engagement with **DRAM** to review their stable coin called **Dram**.

The review was conducted over two weeks, from **August 14, 2023** to **August 18, 2023**, by **Chingiz Mardanov**. A total of 5 person-days were spent.

2 Scope

Our review focused on the commit hash `291600c50c295b3bc473c975d504bc5671d5fabe`. The list of files in scope can be found in the Appendix. We have also conducted the review of fixes on the commit hash `b70348e6998e35282212243ea639d174ced1ef2d`.

After some time **DRAM** team identified that another coin with the same symbol was deployed and have made a decision to change the token symbol from **DRM** to **DRAM**. This update was done in commit `70fc1cd171acf0997f034e50e2ff53f44ef4e452` and does not modify anything else in the contracts that were a part of the audit.

2.1 Objectives

Together with the **DRAM** team, we identified the following priorities for our review:

1. Correctness of the implementation, consistent with the intended functionality and without unintended edge cases.
2. Identify known vulnerabilities particular to smart contract systems, as outlined in our Smart Contract Best Practices, and the Smart Contract Weakness Classification Registry.

3 System Overview

The code in question is a new stable coin that is going to be called Dram and will be primarily backed by the United Arab Emirates currency Dirham. As of right now the stable coin will be minted and initially distributed by the team.

Here are few notable functionalities present in the stable coin's contract:

- **Freezing** - malicious actor's account can be frozen which will prevent Dram from being sent to or from them.
- **Minting and Burning** - DRAM team can mint and burn the tokens as well as offer the ability to mint Dram to different operators up to a certain cap.
- **Pausing** - Dram transfers can be paused completely.

4 Security Specification

This section describes, **from a security perspective**, the expected behavior of the system under audit. It is not a substitute for documentation. The purpose of this section is to identify specific security properties that were validated by the audit team.

4.1 Actors

The relevant actors are listed below with their respective abilities:

- **Role Manager** - account that is responsible for adjusting the mint-caps (amount a specific wallet could mint) as well as granting new roles to other accounts.
- **Regulator** - account that is capable of freezing and unfreezing the holder accounts. This will in turn either pause or unpause transfers of the tokens from and to the frozen account.
- **Supply Manager** - account that is responsible for minting and burning the tokens that were either approved or transferred to it.
- **Admin** - account that is capable of doing all of the above actions.

4.2 Trust Model

It's worth noting that the `Dram` contracts are heavily centralized, and there are some basic assumptions about how the whole system will work that we should talk about.

The key thing to know about these contracts is that they can be upgraded. Depending on how they're set up, the functions of these contracts might change whenever necessary.

Most of the important actions in the contract are controlled by specific roles that manage things like creating or destroying tokens, as well as freezing accounts. It's crucial that these roles are well-protected and not vulnerable to any misuse by malicious parties.

As for the team at `DRAM`, they plan to put most roles, except for the `REGULATORY_MANAGER_ROLE`, behind a timelock contract. This means there's a waiting period before those roles can take action. However, the `REGULATORY_MANAGER_ROLE` won't have this delay since regulators might need to act quickly in emergencies. All these roles will be set up in a way where only the team's multi-signature wallets can propose any changes to the contracts. It's important to mention that while this setup is the plan, we can't guarantee it will always be followed exactly, given the amount of human element involved.

5 Findings

Each issue has an assigned severity:

- Minor issues are subjective in nature. They are typically suggestions around best practices or readability. Code maintainers should use their own judgment as to whether to address such issues.
- Medium issues are objective in nature but are not security vulnerabilities. These should be addressed unless there is a clear reason not to.
- Major issues are security vulnerabilities that may not be directly exploitable or may require certain conditions in order to be exploited. All major issues should be addressed.
- Critical issues are directly exploitable security vulnerabilities that need to be fixed.

5.1 All Roles Are Set to the Same Account. Minor ✓ Fixed

Resolution

All roles, including regulatory manager, are now set to different accounts. The modification can be found in commit `b70348e6998e35282212243ea639d174ced1ef2d`

Description

From talking to the team we know that all roles will be held by different timelock contracts. In the code they all are initiated to the same `admin` address. That would mean that most roles would need to be transferred. Given that each transfer take 2 transactions and there are 3 roles to transfer that would equate to 6 transactions just to properly set up the contract on deployment. That also increments the time it would take and space for making errors.

It is also should be noted that the `regulator` role is not being initialized there at all.

Examples

contracts/access/DramAccessControl.sol:L77-L84

```
// solhint-disable-next-line func-name-mixedcase
function __DramAccessControl_init_unchained(
    address admin
) internal onlyInitializing {
    _grantRole(ADMIN_ROLE, admin);
    _grantRole(ROLE_MANAGER_ROLE, admin);
    _grantRole(SUPPLY_MANAGER_ROLE, admin);
}
```

Recommendation

We suggest passing several addresses into the constructor and setting them to the correct addresses right away. Alternatively one can not set them at all and grant those roles later in order to avoid revoking the roles that admin should not have, such as `SUPPLY_MANAGER_ROLE`.

5.2 Setting MintCap to a Specific Value Is Prone to Front-Running. Minor

Resolution

Acknowledged by the team with a comment: We will modify `mintCaps` using the `increaseMintCap` and `decreaseMintCap` functions.

Description

`Dram` stable coin is using the approval-like model to set the minting caps of different operators, thus it is prone to the same front-run issues as the approval mechanism. When using the `setMintCap` function directly operator could front-run the transaction and completely spend the old cap and then spend the new one again after setting the transaction goes through.

contracts/Dram.sol:L110-L115

```
function setMintCap(
    address operator,
    uint256 amount
) external onlyRoleOrAdmin(ROLE_MANAGER_ROLE) {
    _setMintCap(operator, amount);
}
```

Examples

Imagine the following scenario:

1. Alice has a mint cap of 10.

2. A transaction is sent to the mem-pool to set it to 5 (decrease the cap). The intent is that Alice should only be able to mint 5 tokens.
3. Alice frontruns this transaction and mints 10 tokens.
4. Once transaction 2 goes through Alice mints 5 more tokens.

In total Alice minted 15 tokens.

Recommendation

Avoid using setting the specific mint caps and rather use increase/decrease methods that are present in the code already.

5.3 Context.sol Is Not Required in the Present Use Case. Minor ✓ Fixed

Resolution
Fixed in commit: `b70348e6998e35282212243ea639d174ced1ef2d`

Description

`ContextUpgradeable.sol` contract that is used in the `DramMintable` contract can be regarded as a foundational contract, the methods of which are intended for overriding to facilitate the implementation of varying meta transaction logics.

contracts/token/ERC20/extensions/DramMintable.sol:L18-L23

```
abstract contract DramMintable is
    Initializable,
    IDramMintable,
    ContextUpgradeable,
    ERC20Upgradeable
{
```

In its isolated state, it brings no additional value but rather makes code evaluation more difficult.

Recommendation

In the case of `DramMintable` `msg.sender` can be used directly. In the case where meta-transactions are planned to be utilized, `ContextUpgradeable.sol` has to be overridden accordingly.

5.4 Admin Can Mint and Burn Tokens Which Is Not Immediately Evident From Code.

Description

Burning and minting tokens functions both have the following modifier: `onlyRole(SUPPLY_MANAGER_ROLE)` which is different from `onlyRoleOrAdmin` used elsewhere. That implies that the `Admin` can not do minting or burning while in reality `Admin` can do that by just granting themselves a supply manager role first.

Examples

contracts/Dram.sol:L126

```
) external onlyRole(SUPPLY_MANAGER_ROLE) {
```

contracts/Dram.sol:L135

```
function burn(uint256 amount) external onlyRole(SUPPLY_MANAGER_ROLE) {
```

Appendix 1 - Files in Scope

This audit covered the following files:

File	SHA-1 hash
contracts/Dram.sol	`c082e30f7ab6c4412119d8a0b1cf7a46464fd16a`
contracts/access/DramAccessControl.sol	`eb6416f5c837f6b4a6b8e3e23552aafab5b506b0`
contracts/access/DramFreezable.sol	`7648a6545652adf32fb68a0ed09ea2df0f3c0a6a`
contracts/access/IDramAccessControl.sol	`882b9c60caa490b556ce451158bce31ef4e5321f`
contracts/access/IDramFreezable.sol	`7abccaca6c15103814651080a4f9b22fa2a7c45c`
contracts/token/ERC20/extensions/DramMintable.sol	`834270ea9c48dc6e82fddc05acdcc32ea5d93391`
contracts/token/ERC20/extensions/IDramMintable.sol	`70578fe21c5586fa222c6ed49635e5fa81d7427c`

Appendix 2 - Disclosure

Consensys Diligence (“CD”) typically receives compensation from one or more clients (the “Clients”) for performing the analysis contained in these reports (the “Reports”). The Reports may be distributed through other means, including via Consensys publications and other distributions.

The Reports are not an endorsement or indictment of any particular project or team, and the Reports do not guarantee the security of any particular project. This Report does not consider, and should not be interpreted as considering or having any

bearing on, the potential economics of a token, token sale or any other product, service or other asset. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. No Report provides any warranty or representation to any third party in any respect, including regarding the bug-free nature of code, the business model or proprietors of any such business model, and the legal compliance of any such business. No third party should rely on the Reports in any way, including for the purpose of making any decisions to buy or sell any token, product, service or other asset. Specifically, for the avoidance of doubt, this Report does not constitute investment advice, is not intended to be relied upon as investment advice, is not an endorsement of this project or team, and it is not a guarantee as to the absolute security of the project. CD owes no duty to any third party by virtue of publishing these Reports.

A.2.1 Purpose of Reports

The Reports and the analysis described therein are created solely for Clients and published with their consent. The scope of our review is limited to a review of code and only the code we note as being within the scope of our review within this report. Any Solidity code itself presents unique and unquantifiable risks as the Solidity language itself remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond specified code that could present security risks. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. In some instances, we may perform penetration testing or infrastructure assessments depending on the scope of the particular engagement.

CD makes the Reports available to parties other than the Clients (i.e., “third parties”) on its website. CD hopes that by making these analyses publicly available, it can help the blockchain ecosystem develop technical best practices in this rapidly evolving area of innovation.

A.2.2 Links to Other Web Sites from This Web Site

You may, through hypertext or other computer links, gain access to web sites operated by persons other than Consensys and CD. Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites’ owners. You agree that Consensys and CD are not responsible for the content or operation of such Web sites, and that Consensys and CD shall have no liability to you or any other person or entity for the use of third party Web sites. Except as described below, a hyperlink from this web Site to another web site does not imply or mean that Consensys and CD endorses the content on that Web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the Reports. Consensys and CD assumes no responsibility for the use of third-party software on the Web Site and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any outcome generated by such software.

A.2.3 Timeliness of Content

The content contained in the Reports is current as of the date appearing on the Report and is subject to change without notice unless indicated otherwise, by Consensys and CD.