Scram Platform Security

Jeffrey I. Schiller*

January 22, 2021

1 Abstract

The Secure Cyber Risk Aggregation and Measurement (SCRAM) platform implements a secure platform for aggregating sensitive information. This paper gives an overview of how this platform is implemented, explaining the use of algorithms and the system that provides its security.

2 Introduction

The intended purpose of the SCRAM platform is to provide a means for organizations and individuals to supply sensitive information that will be used to compute an aggregate function. For example, it could be used by a group of individuals who wish to know the average salary of the group without any individual having to reveal their salary to anyone else.

The SCRAM platform provides security through both traditional computer/network security techniques as well as through a novel cryptographic approach that provides an important extra level of protection, including from those who operate the platform.

3 Platform description

The SCRAM Platform conforms to a client/server architecture. The client portion is run in a web browser using Javascript and Webassembly (WASM). [1]

The server is a Linux computer(s) which contains a web server, specialized programs for performing the SCRAM computations and a database to store results. It is worth noting now that no sensitive data is ever uploaded to the server or stored in the corresponding database.

*jis@mit.edu

[1] Webassembly is a new client side binary technology implemented in a VM in modern web browsers. You can read more about it at https://webassembly.org/.

4 The SCRAM Cryptographic Approach

In this section we give a simple over-view of the cryptographic methods used by SCRAM. However, we do not go into details on the mathematics that underpin it. For that, please see *SCRAM: A Platform for Securely Measuring Cyber Risk*[Cas+20].

The goal of the cryptographic approach is to ensure that input data to whatever computation is desired is never revealed to anyone other then the entity that provided that input. This includes protection against those who operate the server which is gathering the input.

The SCRAM algorithm takes input data, encrypts it and then sends it to a central server. The keys needed to decrypt the input are distributed among a group of key holders who have to all contribute to any decryption. In normal operation, input data is never decrypted. Instead it is combined to produce an aggregate computation, for example an average. This computation works on the encrypted data, thus providing confidentiality of the origin input.

4.1 Participant Roles

The SCRAM system has three types of participant roles. A given participant can have one or more roles assigned to them. They are:

4.1.1 Data Entry Role

The "Data Entry Role," as its name implies is the role where a participant has data to input into the aggregate computation.

4.1.2 Key Holder Role

Perhaps the most important role when it comes to ensuring the security of the computation. The key holder role creates and uses a public-private key pair.[2] The public keys of all of the key holders are combined to create a group public key which is used to encrypt each individual data input. All the private keys, held by each of the keyholders, are required to decrypt any submitted data or the results. The distributed nature of the keys provides a highly secure environment.

4.1.3 The Admin Role

The Admin role runs the computation. In particular the Admin Role determines when it is time to combine the public keys of the key holders. Once this is done, no further key holders may join the computation. The Admin role also combines the encrypted inputs together to form the aggregate (but still encrypted) result. Finally after all of the key holders have performed decryption, the Admin combines their results to arrive at the decrypted aggregate computation.

[2] In traditional cryptography, a single key is used to encrypt and then decrypt data. In a public key system, a pair of keys are created the public key is used in one operation, encryption in our case, and the private key is used for the other operation, decryption in our case. The defining characteristic of a public-private key system is that the private key can not feasibly be recovered given only the public key.

4.2 How a computation works

The first step in setting up a computation is for a person with the Admin role to create the computation. This involves defining the aggregate functions to be performed and the identification of the input fields.[3]

Once the computation is setup, participants configured as key holders need to login and generate their public-private key pair.

After all of the keys are generated, the admin role combines the public keys of the key holders into a combined group public key.

The participants who have data to enter at this point can enter data, which is encrypted and stored on the server.

Once all of the data is entered, the admin combines the encrypted data to compute the desired aggregate function.

The key holders now each need to login and using their private key, they do a partial decryption of the combined encrypted aggregate. Once all key holders have contributed their partial decryption, the admin role combines the decrypted shares of the computation to reveal the plain text of the computed aggregate.

As can be seen, performing a computation is a multi-step process. For participants who are simply providing data for the computation, the process is fairly simple. Once the combined public key is created, all they need to do is enter data.

The key holders, however, have a more complex role. Each key holder must create their public-private key pair before any data can be input. Once the aggregate computation is performed, each key holder must login and perform their portion of the decryption. If a single key holder fails to decrypt their portion, the entire computation is lost and needs to be redone.

4.3 Security of the various steps

The goal of the SCRAM platform is to perform all steps of a computation securely. Below we discuss how this is accomplished.

4.3.1 Generating key holder keys

When a key holder generates a key pair, WASM code is downloaded from the server to their browser. The browser then creates the key pair. The user is prompted for a pass phrase which is used to generate an encryption key which is then used to encrypt the private key. The key holder has the option of either storing the encrypted private key on the SCRAM server, or in a file on their own computer. If they choose the later, the private key is **not** uploaded to the SCRAM server.

4.3.2 Encrypting Input Data

When data is input, it is not simply added to a web form. Instead the client (browser) downloads WASM code and the combined group public key. It then takes the data entered and encrypts it with the public key and uploads the encrypted result to the server.

[3] A given computation can have multiple values which are independently operated on, using the same keys

4.3.3 Aggregate Decryption

Once all data is entered, the admin role combines the input data to generate the encrypted aggregate result. The key holders then each login and use their private key to perform a partial decryption. Like in the previous cases, the encrypted data is downloaded to their browser along with the WASM code to do the private key decryption. If they chose to store their private key on the SCRAM server, then it too is downloaded to the browser. The WASM code in the browser now performs the partial decryption and uploads the result to the SCRAM server.

4.3.4 Generating the Final Result

Once all of the partial decryptions are uploaded to the SCRAM server, the admin role can perform the final computation on the server which reveals the unencrypted aggregate result.

4.4 Security vs. Reliability Trade-Off

There is a trade-off to be made between the number of key holders and the over-all security of a computation. The more key holders, the more difficult it will be for an intruder to compromise a computation. However the more key holders there are, the more difficult is the coordination of them to perform the steps of the computation that they need to be involved in. Also, as the number of key holders increases, so does the risk that a key holder will lose access to their private key either by losing a key file they stored on their own computer, or forgetting the password used to protect it.

5 Over-all platform security

In addition to the security provided by the SCRAM encryption algorithm, the server platform is itself securely maintained. All communication between the web browser(s) and the server are encrypted with TLS (aka https) even though most of the communication is not sensitive.

The server itself has no account with passwords, and the root account cannot be directly logged into. All administrative access is via SSH with public key authentication.

The SCRAM encryption algorithm ensures that a read-only compromise of the SCRAM server will never reveal any of the sensitive data input. However a compromise of the server that permits the intruder to alter the SCRAM software, in particular the WASM software that is downloaded to client browsers can compromise the platform.

We take additional steps to address this threat. In addition to the normal security precautions taken to manage the server, there is another set of servers that remotely monitor the server. About every 5 minutes, code is downloaded to the server, which resides only in memory, that computes a cryptographic checksum of the critical SCRAM software components. This will detect any authorized changes with very high probability.

Another threat beyond the control of the SCRAM platform is the protection of the private keys of the key holders. It is the responsibility of each key holder to use a strong password to protect their private key. If they chose to store their private key on their own computers, they need to protect it there as well. It is important that key holders not lose their keys, as that makes it impossible to complete a computation.

However, mitigating the threat to key holders is the fact that only by compromising all of the key holder private keys is any data at risk of unauthorized disclosure. In security we like to say that the strength of

the system is like a chain, it is only as strong as its weakest link. However in SCRAM, the strength of the key holder mechanism is as strong as the strongest key holder.

In order to compromise input data to a computation an intruder would need not only all of the key holder private keys, but also privileged access to the SCRAM server in order to obtain the raw encrypted data from participants.

5 Summary

The SCRAM platform offers a secure way of computing aggregate computations over sensitive data. The security of the over-all system is provided not only by traditional server security mechanisms, but also by the novel SCRAM algorithm and architecture that distributes critical information (keys) to a range of parties.

References

[Cas+20] Leo de Castro et al. "SCRAM: A Platform for Securely Measuring Cyber Risk". In: *Harvard Data Science Review* (Sept. 16, 2020). https://hdsr.mitpress.mit.edu/pub/gylaxji4. DOI: 10.1162/99608f92.b4bb506a. URL: https://hdsr.mitpress.mit.edu/pub/gylaxji4.