

GSM Security as Service on Cloud

Introduction:

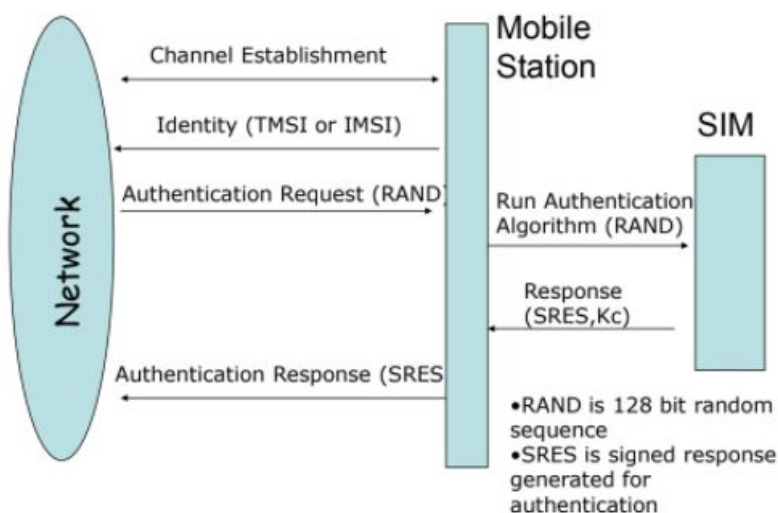
The GSM (Global System for Mobile Communications) is a widely used cellular standard in the world. Recently, the mobile industry has experienced an extreme increase in the number of its users. Thus, the GSM network with the greatest worldwide number of users succumbs to several security vulnerabilities. Security is a burning and intelligent issue. GSM security flaws have been identified several years ago. Many algorithms are used for making the GSM secure. The algorithms mainly used are A3, A5, and A8 algorithms. Algorithm A3 is used for authentication, A5 is used for encryption, and A8 is used for the generation of a cipher key. This paper presents an enhanced scheme of the A3 algorithm to improve the level of security provided by the GSM.

The GSM Security Model is based on a COMP128 authentication algorithm between the subscriber's home network's and the subscriber's phone card, it is used to provide secure identification of the subscriber on the GSM networks. Eventually, the GSM algorithms leaked out and have been studied extensively ever since by the open scientific community. Each SIM card in its GSM handset contains a different 128-bit secret key K, known only to it and to an Authentication Centre (AuC) in the network.

Elastic Beanstalk supports applications developed in Java, PHP, .NET, Node.js, Python, and Ruby, as well as different container types for each language. A container defines the infrastructure and software stack to be used for a given environment. Elastic Beanstalk provides developers and systems administrators an easy, fast way to deploy and manage their applications without having to worry about AWS infrastructure. Elastic Beanstalk does not restrict your choice of persistent storage and database service options.

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. The basic building block of Amazon RDS is the *DB instance*. A DB instance is an isolated database environment in the cloud. A DB instance can contain multiple user-created databases, and you can access it by using the same tools and applications that you use with a stand-alone database instance.

1. GSM Authentication - A3 and A8 Algorithm:



When a GSM phone call is being set up the AuC sends RAND, a freshly generated 128-bit random number, via the network and the GSM handset to the SIM. The SIM combines RAND and K using a hash function called A3, which gives a 32 bit “signed response” SRES. The SIM sends SRES back to the AuC via the handset and the network.

The AuC compares SRES to the value which is computed using its own copy of RAND and K. If they are equal, the AuC believes that the SIM is authentic and the call is allowed to proceed. The speech exchanged between the GSM handset and the network is encrypted using an algorithm called A5 which has a 64-bit session key. For each new call, the required A5 session key is generated using a hash function called A8. This takes the same 128-bit challenge and 128-bit key K and produces the 64-bit session key, so no further exchange of data is required for this step.

- i. The MS will send either an IMSI or a TMSI to the BSS.
- ii. The BSS forwards the MSC/VLR
- iii. The MSC/VLR forwards the IMSI to the HLR and requests verification of the IMSI as well as Authentication Triplets
- iv. The HLR will forward the IMSI to the Authentication Center (AUC) and request authentication triplets
- v. The AUC generates the triplets and sends them along with the IMSI, back to the HLR
- vi. The HLR validates the IMSI by ensuring it is allowed on the network and is allowed subscriber services. It then forwards the IMSI and Triplets to the MSC/VLR
- vii. The MSC/VLR stores the SRES and the Kc and forwards the RAND to the BSS and orders the BSS to authenticate the MS
- viii. The MS uses the RAND to calculate the SRES and sends the SRES back to the BSS
- ix. The BSS forwards the SRES up to the MSC/VLR
- x. The MSC/VLR compares the SRES generated by the AUC with the SRES generated by the MS. If they match, then authentication is completed successfully

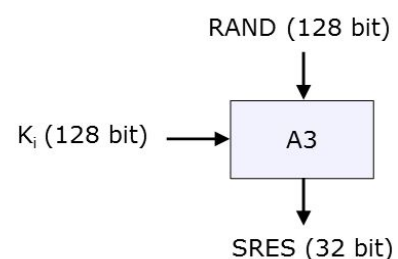
A3 - Authentication:

A3 Input:

- 128-bit RAND random
- Ki 128-bit private key

A3 Output:

- 32-bit SRES signed response



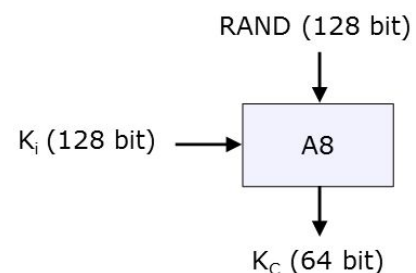
A8 - Key Generator

A8 Input:

- 128-bit RAND random
- Ki 128-bit private key

A8 Output:

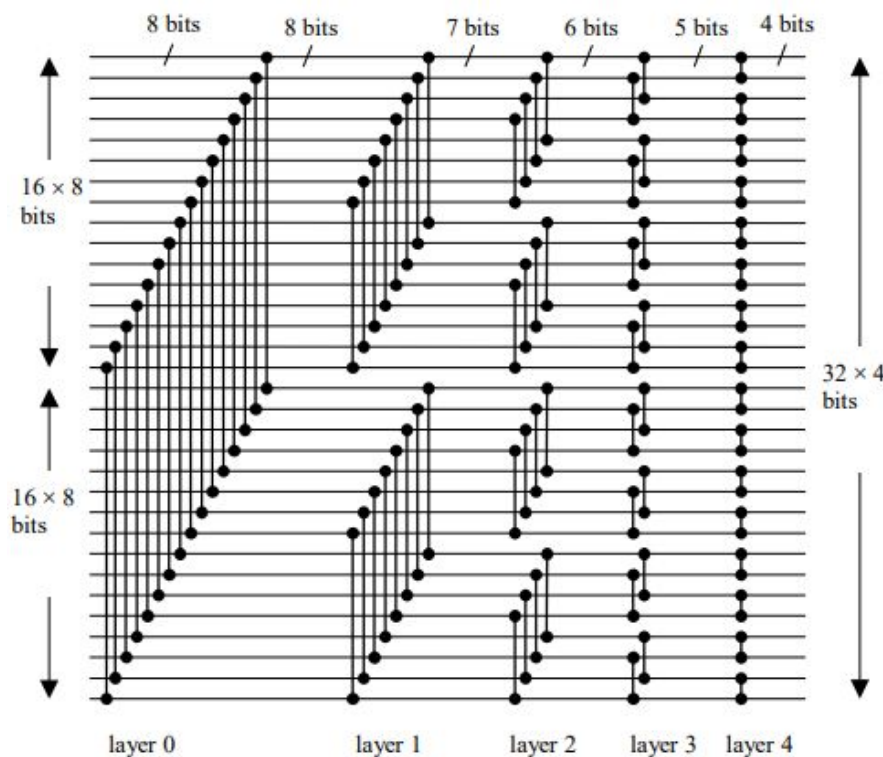
- 64-bit Kc Cipher Key



COMP128

The **COMP128** algorithms are implementations of the A3 and A8 algorithms defined in the GSM standard. The A3 algorithm is used to authenticate the mobile station to the network. The A8 algorithm is used to generate the session key used by A5 to encrypt the data transmitted between the mobile station and the BTS.

COMP128 design was completely private. The algorithm was not released to the public, thus it lacks much-needed peer review. Currently, there exist four versions of COMP128. The first three were originally confidential. A partial description of the first version was leaked in 1997 and completed via reverse engineering. This led to a full publication in 1998. The second and third versions were obtained via reverse engineering of software which verifies SIM cards compliance.



This is commonly called 'Butterfly Structure.' In each of the 5 levels, compression is performed on 2 equal-sized sections. E.g. In round 0, 2-16 byte sections, round 1, 4-8 bit sections, etc. For level 'i', T_i (table) contains $29-i$ (8-i)-bit values. e.g. T_0 has 512 8-bit values, while T_4 has 32 4-bit values. In each level, two input bytes are used to calculate the index for the table and the result is the output byte.

2. Amazon Web Services:

The AWS Cloud provides a broad set of infrastructure services, such as computing power, storage options, networking and databases that are delivered as a utility: on-demand, available in seconds, with pay-as-you-go pricing. From data warehousing to deployment tools, directories to content delivery, over 90 AWS services are available. New services can be provisioned quickly, without upfront capital expense. This allows enterprises, start-ups, small and

medium-sized businesses, and customers in the public sector to access the building blocks they need to respond quickly to changing business requirements. This whitepaper provides you with an overview of the benefits of the AWS Cloud and introduces you to the services that make up the platform.

3.1 Amazon Relational Database Service (Amazon RDS)

Amazon RDS is a managed relational database service that provides you six familiar database engines to choose from, including Amazon Aurora, MySQL, MariaDB, Oracle, Microsoft SQL Server, and PostgreSQL. This means that the code, applications, and tools you already use today with your existing databases can be used with Amazon RDS. Amazon RDS handles routine database tasks such as provisioning, patching, backup, recovery, failure detection, and repair.

Amazon RDS makes it easy to use replication to enhance availability and reliability for production workloads. Using the Multi-AZ deployment option, you can run mission-critical workloads with high availability and built-in automated fail-over from your primary database to a synchronously replicated secondary database. Using Read Replicas, you can scale out beyond the capacity of a single database deployment for read-heavy database workloads.

MySQL is the world's most popular open-source relational database and Amazon RDS makes it easy to set up, operate, and scale MySQL deployments in the cloud.

Benefits of using Amazon RDS:

- **Easy To Administer**
You can use the AWS Management Console, the Amazon RDS Command Line Interface, or simple API calls to access the capabilities of a production-ready relational database in minutes. Amazon RDS database instances are pre-configured with parameters and settings appropriate for the engine and class you have selected. You can launch a database instance and connect your application within minutes.
- **Scalability**
You can scale the compute and memory resources powering your deployment up or down, up to a maximum of 32 vCPUs and 244 GiB of RAM. When using the Provisioned IOPS and General Purpose (SSD) storage types, you can create MySQL instances with up to 16 TiB of storage.
- **Availability and Durability**
When you provision a Multi-AZ DB Instance, Amazon RDS automatically creates a primary DB Instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ). Amazon RDS has many other features that enhance reliability for critical production databases, including automated backups, database snapshots, and automatic host replacement.
- **Fast**
Amazon RDS supports the most demanding database applications.
- **Secure**
Amazon RDS makes it easy to control network access to your database. Amazon RDS also lets you run your database instances in Amazon Virtual Private Cloud (Amazon VPC), which enables you to isolate your database instances and to connect to your existing IT infrastructure

3.2 Elastic Beanstalk

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.

There is no additional charge for Elastic Beanstalk - you pay only for the AWS resources needed to store and run your applications. Elastic Beanstalk for PHP makes it easy to deploy, manage, and scale your PHP web applications using Amazon Web Services.

An Elastic Beanstalk application is a logical collection of Elastic Beanstalk components, including environments, versions, and environment configurations. In Elastic Beanstalk an application is conceptually similar to a folder.

Elastic Beanstalk provides configuration options that you can use to customize the software that runs on the EC2 instances in your Elastic Beanstalk environment. You can configure environment variables needed by your application, enable log rotation to Amazon S3, and set common PHP initialization settings.

Platform-specific configuration options are available in the AWS Management Console for modifying the configuration of a running environment. To avoid losing your environment's configuration when you terminate it, you can use saved configurations to save your settings and later apply them to another environment.

Benefits of using Elastic Beanstalk:

- **Fast and simple to begin**
Elastic Beanstalk is the fastest and simplest way to deploy your application on AWS. You simply use the AWS Management Console, a Git repository to upload your application, and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing and auto-scaling
- **Developer productivity**
Elastic Beanstalk provisions and operates the infrastructure and manages the application stack (platform) for you, so you don't have to spend the time or develop the expertise.
- **Impossible to outgrow**
Elastic Beanstalk automatically scales your application up and down based on your application's specific need using easily adjustable Auto Scaling settings.
- **Complete resource control**
You have the freedom to select the AWS resources, such as Amazon EC2 instance type, that are optimal for your application. Additionally, Elastic Beanstalk lets you "open the hood" and retain full control over the AWS resources powering your application.

There are 3 main elements involved in the GSM Authentication Process:

1. Subscriber Identity Module(SIM)
2. Mobile Service Switching Center(MSC)
3. Authentication Center(AUC)

In our cloud implementation of GSM Authentication, a web browser acts like a SIM and PHP web server created using Elastic Beanstalk acts as an MSC and AUC. We created one Amazon RDS instance running the MYSQL server. It acts as a database of AUC. This database contains a list of IMSIs and their corresponding 128-bit secret keys. GSM authentication on cloud occurs in the following steps:

1. The user enters the mobile number in the input box on web page and clicks authenticate button.
2. After clicking the button, mobile number is sent to the web server using ajax POST request. Web server searches for an entry corresponding to sent mobile number. If entry is not found, the message "Invalid SIM" is displayed.
3. Else, Ki for that mobile number is retrieved. One PHP function on the web server generates RAND. SRES and Kc are calculated using the COMP128 algorithm implemented in PHP. SRES is stored as a session variable to make it available when we compare it later with RES computed by the browser.
4. RAND and SRES are then sent to the browser as a response to the POST request. Browser, then, looks for the entry of the mobile number in sims.json file which acts as a collection of sim configurations. If entry is not found, "SIM unavailable" message is displayed.
5. Else RES is calculated on the browser side using the COMP128 algorithm implemented in Javascript. RES is sent to a web server using ajax POST request.
6. SRES stored in session variable and RES are compared. If they match "Authentication Successful" message is displayed. Else, "Authentication Failed" message is displayed.


← → ↻ ⓘ Not secure | gsmsecurity-env.us-east-2.elasticbeanstalk.com ☆ ⋮

GSM Security

Mobile Number:

+91 8779156986

Authenticate

 **SIM**


Requesting for Authentication

Received RAND...

Generating RES...

RES:00000000000000000000111111111111

RES sent to MSC

 **MSC**

Request Received...

Challenge Generated...

RAND:
[249,213,135,15,39,202,27,32,140,157,132,167,11,82,179,42]
Kc:00000011111111111111111111111111000000000000000011111000000
SRES:00000000000000000000111111111111

RES received from SIM

SRES and RES match

Successfully Authenticated