

1

Introduction

Over the last few decades, information technology has changed the world in two major ways. The first development is computerization. Computers have gradually entered into almost every walk of life. Many services that used to require interaction with humans are now provided by computers. When you withdraw money from your bank account, or look for a book in the library, or check in at the airport, chances are that you are interacting with computer systems hosting these services.

The second development is the immense and increasing popularity of an open network of interconnecting computers, the Internet. Access to computerized and automated services now takes place over this open network: when you use online banking to make payments from your bank account, or order books from an online bookstore, or browse the online photo albums of your friends, you are using the Internet, as well as a variety of communication links to access the Internet.

In this chapter, we will discuss what is needed to secure the access to applications and services over open networks, what the primary difficulty in achieving security is, and why Generic Authentication Architecture (GAA) helps solve this problem. We will also outline how this book is structured and how to go about reading it.

1.1 Authenticated Key Agreement

Although the nature of accessing services has changed, many of these services require some form of controlled access: for example, only you should be able

to make payments from your bank account. Usually, controlling access to services is contingent on identifying who is requesting access and verifying the requestor's identity. In other words, the serving computer has to *authenticate* the requestor.

In a closed network, like the plain old telephone system, authentication can be implicit based on the presumed physical security of the network. But in an open network like the Internet, physical security is not relevant – it is easy to claim any identity towards a distant server. Worse still, it is easy to pretend to be a distant server towards an unsuspecting client (for example, using IP address spoofing). Therefore, we need to make use of cryptographic techniques for *mutual authentication* in order to have sufficient trust in the authentication process.

In open networks, authenticating the parties at the beginning of a communication session is not sufficient: An attacker may wait for authentication to complete and then hijack the session by inserting, modifying or deleting the messages being exchanged. To prevent this, the authentication process should also establish *session keys* which can be used to guarantee the integrity of the entire communication session.

In some services, the messages exchanged may need to be private. For example, suppose you have an online photo album accessible only to family and friends. When a friend is legitimately viewing the pictures on your album, the information has to travel from the album server to your friend's browser. It may traverse several communication links with varying levels of physical security. For example, your friend's computer may be connected to her access router over an open wireless link. You do not want anyone eavesdropping along the way to be able to see your pictures. Cryptographic techniques for encryption can protect the messages while en route. Session keys established during the initial authentication can be used for encrypting messages exchanged during the session.

The process of mutual authentication and session key agreement is known as *authenticated key agreement*.

1.2 The Challenge in Authenticated Key Agreement

Mechanisms and protocols for performing authenticated key agreement are well known. For example, every time you access a protected web server, your browser and the web server engage in the Transport Layer Security (TLS) handshake protocol for authenticating the server and agreeing on a session key. The challenge in the authentication is in the task of initializing the necessary credentials at the parties involved in the authentication.

Consider what typically happens when you enrol into the authentication system in the bank. You have to visit the bank in person, and possibly show some photo identification to open an account and provide a mailing address. The bank will then

send you the credentials needed to access the bank account, for example, a bank card and the personal identification number (PIN) in separate mails. The process costs time and money. One approach to reduce the cost of initialization is to relax the expected level of security and usability. This is the approach taken by popular free e-mail services: initialization is done by the user visiting a signup page and choosing a username and setting a password. The user ends up using the same password for many different services or puts up with the inconvenience of remembering many different passwords. This may in turn cause that users are more frequently calling the help desk, or a special password recovery tool is required, which also introduces costs to the service provider.

An alternative approach is to bootstrap the needed credentials from an existing security infrastructure. One such security infrastructure is the cellular security infrastructure. The cellular security infrastructure has several characteristics that make it a particularly attractive infrastructure to bootstrap security for applications from.

The first and foremost characteristic is its scalability. The Global System for Mobile Communications (GSM) and Universal Mobile Telecommunications System (UMTS) infrastructure consists of hundreds of participating mobile operators and over two billion subscribers worldwide. Most mobile operators have roaming and billing agreements with many other operators. Once you enrol as a GSM / UMTS subscriber with a local operator in your home country, you will be able to authenticate to many mobile operators, and use their networks to make phone calls or send and receive messages.

The second characteristic is its ease-of-use. Cellular authentication is an example of security that remains under the hood and just works. Users are not required to perform any verification or understand technical security concepts.

The third characteristic is its level of security. Authentication in cellular systems is based on the possession of smart cards. Even though some of the cryptographic algorithms in earlier versions of GSM have been broken, the entire system has stood the test of time. GSM / UMTS security architecture is beginning to be acknowledged as an example of the principle of ‘good enough’ security of striking the right balance among cost-effectiveness, security and usability [Sandhu03].

GAA consists of a set of specifications that describe how the cellular security infrastructure can be used to provide a general-purpose authentication service for applications and services. It has been standardized both in the 3rd Generation Partnership Project (3GPP) and its North American counterpart the 3rd Generation Partnership Project 2 (3GPP2). Deployment of GAA in mobile devices and mobile networks is expected to start in 2008.

The GAA concept is illustrated in Figure 1.1. GAA is a generic architecture for mutual authentication and key agreement (AKA). Its fundamental building block is Generic Bootstrapping Architecture (GBA). GBA enables automatic provisioning of

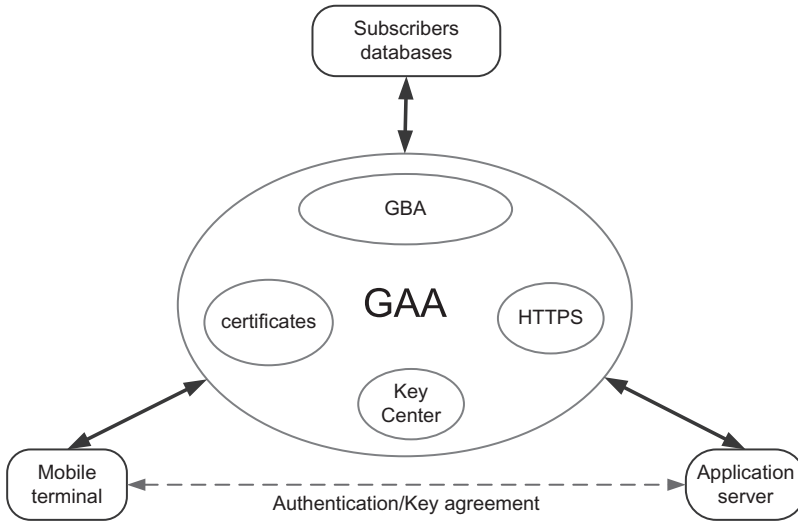


Figure 1.1. Generic Authentication Architecture (GAA) concept

shared keys between the mobile terminal and an application server (provided that the user has a valid subscription¹ to cellular network services).

Other GAA constituents are built on top of GBA (new GAA building blocks continue to be specified):

- Support for subscriber **certificates** (SSC) specifies procedures for the registration of user's public keys. Those procedures are authenticated with GBA.
- Access to application servers with **HTTPS** specifies how to use shared keys created with GBA in conjunction with server-authenticated HTTPS to establish secure and mutually authenticated HTTP communication between mobile terminal and application server.
- **Key Centre** enables creation of keys shared between terminals.

Broadcast mobile television and Multimedia Broadcast/Multicast Service (MBMS), in which encrypted content is wirelessly broadcast, or multicast, are the lead applications driving the deployment of GAA. In those applications, the delivery of service keys that are needed to decrypt the received content is secured with GAA. This makes valid cellular subscription a prerequisite for, e.g., watching mobile television programs. Also, when generating charges for mobile television and MBMS, the person's cellular identity, which is verified with GAA, is used.

¹Valid subscription implies that the mobile terminal and subscriber's databases have a copy of shared key that is used in cellular authentication.

1.3 How to Read this Book?

Our goal in writing this book is to explain what GAA is and how it can be used. We have four different types of readers in mind:

- **Developers** are software designers who design and implement new applications and services. We show how developers can make their application software use GAA for authentication.
- **Architects** are technical experts who design protocols and systems. We explain the GAA concepts and technical details and show examples of how GAA is integrated into existing protocols so that architects can determine if and how GAA could be used to solve the authentication needs of the systems they are designing.
- **Executives** are decision makers in companies who need to figure out whether need to deploy GAA. We provide a general overview of GAA and brief analyses of its benefits and tradeoffs that can serve as background materials for decision making.
- **Academics** are university professors, researchers and students studying computer science or communication systems. We explain the principles and technical details in the design of GAA that can serve as starting points for academics interested in analyzing and evaluating GAA, comparing it to other authentication systems, and designing authentication systems for the future.

The chapters are arranged in the logical order in which we recommend the reader to proceed. Table 1.1 indicates which sections are likely to be of interest to different types of readers.

Table 1.1. How to read this book

Section	Developer	Architect	Executive	Academic
1. Introduction				
1.1 Authenticated Key Agreement	✓	✓	✓	✓
1.2 The Challenge in Authenticated Key Agreement	✓	✓	✓	✓
1.3 How to Read this Book?	✓	✓	✓	✓
2. Classical Approaches				
2.1 Existing Mobile Security Solutions		✓	✓	✓
2.2 General-Purpose Approaches		✓		✓
2.3 Requirements for GAA	✓	✓	✓	✓

Table 1.1. (continued)

Section	Developer	Architect	Executive	Academic
3. Generic Authentication				
Architecture (GAA)				
3.1 Overview of GAA	✓	✓	✓	✓
3.2 Foundations of GAA – Generic Bootstrapping Architecture (GBA)	✓	✓		✓
3.3 Variations of GBA		✓		✓
3.4 Building Blocks of GAA		✓		✓
3.5 Other Architectural Issues		✓		✓
3.6 Overview of 3GPP GAA Specifications	✓	✓		
4. Applications Using GAA				
4.1 Standardized Usage Scenarios (incl. Broadcast Mobile TV)	✓	✓	✓	
4.2 Additional Usage Scenarios	✓	✓	✓	
5. Guidance for Deploying GAA				
5.1 Integration with Application Servers	✓	✓		
5.2 Integration with OS Security	✓	✓		
5.3 Integration with ID Management Systems	✓	✓	✓	✓
5.4 Integration of GAA into Mobile Networks		✓		
6. Future Trends	✓	✓	✓	✓
Terminology and Abbreviations	✓	✓	✓	✓

Reference

- [Sandhu03] Ravi S. Sandhu: *Good-Enough Security: Toward a Pragmatic Business-Driven Discipline*. IEEE Internet Computing 7(1): 66–68 (2003).