DEKRA

Whitepaper

From SMS to SIM: Security Analysis of Phone Number Verification

About DEKRA Digital & Product Solutions

Innovating safety and security by creating the intelligent testing solutions that contribute to making the digitalized and connected world a safer place. We are the experts in testing and certifying products and new digital technologies.

We deliver solutions from Cybersecurity, Artificial Intelligence, Big Data, to Connectivity, Product Safety, Electromagnetic Compatibility & Radiofrequency, Product Certification, Medical Devices and Automotive Testing.

Through a global network of 48 state-of-the-art test laboratories and facilities, we offer a broad portfolio of product testing services based on national and international standards as well as industry and customer requirements.

Would you like more information?

Authors

Anders Olof Möller
anders.olofmoller@dekra.com

Fernando González Trigo
fernando.gonzaleztrigo@dekra.com

Contact

Anders Olof Möller
anders.olofmoller@dekra.com

Jorge Carrera Borrego
jorgel.carrera@dekra.com

Contact

Research sponsored by Google

Executive Summary & Outline

In today's digital landscape, where cyber threats and fraud are increasingly sophisticated and widespread, the importance of secure authentication methods cannot be overstated. Fraudsters exploit vulnerabilities in widely-used verification systems, such as SMS-based OTPs, through techniques like phishing, SIM swapping, and attacks using malicious applications. These threats pose significant risks to individuals, companies, and financial institutions, where fraud can lead to severe financial losses and additionally compromise sensitive customer data.

This paper addresses these challenges by evaluating the strengths and weaknesses of various phone number verification technologies. It provides a comprehensive security analysis of the current SMS-OTP phone number verification method, the emerging network-based phone number verification method and the more secure SIM-based option that is under consideration for adoption. The insights and proofs of concept presented in this paper serve as high-level security guidance for the selection and implementation of phone number verification methods, ensuring robust security measures that can withstand the evolving tactics of cybercriminals.

Phone number verification is a security measure used to confirm the identity of users by linking their account or service access to a unique phone number. This process involves for example sending a code to the user's phone, which they must then enter to complete the verification. By ensuring that the user has access to the specified mobile device, phone number verification helps to enhance security, prevent fraud, and protect sensitive information across various online platforms and services, including financial services, during for example account creation, login and transactions.

Behind the concept of phone number verification, there are different methods with their respective strengths and weaknesses. This paper explores SMS-OTP (One-Time Password in Short Message System), Network-based and SIM-based phone number verification for authentication, all based on leveraging security from the mobile network.

A comparative approach is taken and the main security principles are analyzed and illustrated. Based on this, a threat analysis was performed, followed by a review of publicly disclosed attacks. To demonstrate the practicality of real-world attacks, a series of Proofs of Concept (PoCs) were performed. Furthermore, the paper briefly discusses the Payment Services Directive 2 (PSD2) regulatory standard, which mandates Strong Customer Authentication (SCA), where phone number verification is one of the possible authentication methods.

The study concludes that while SMS-OTP is widely adopted, it could be vulnerable to, for example, SMS phishing, attacks on the SMS channel and attacks from malicious applications with access to SMS. On the other hand, network-based phone number verification is effective, user-friendly and mitigates common SMS phishing attacks. However, it seems to be vulnerable to attacks by malicious applications requiring only standard privileges. Consequently, SIM-based verification emerges as a potentially more secure future option.

Table of contents

1 Introduction

Authentication is the process of verifying a user's identity, "that you are you", and that it is not someone else pretending to be you. To perform authentication over a network, different authentication factors and methods can be used.

Authentication factors are typically categorized as:

- Something that you know, such as a password or a key.
- Something that you are, such as biometric factors.
- Something that you have, which could be a physical token or your mobile phone.

Typical use cases where authentication is required include accessing services and resources through a web interface or a local application on a mobile device. Examples of such use cases are:

- A financial service, such as a banking application.
- A communication service, such as email or messaging.
- A commercial service, such as online shopping.
- A community or public service.

Examples of typical user actions that require authentication are:

- User account creation and registration for a service.
- User login to a service and resources.
- User updates of account information, including information related to security, for example changing password or phone number.
- Transactions, for example in banking.

Apart from the widespread use of passwords [1], other examples of authentication solutions include dedicated devices for multi factor authentication, dedicated authentication applications such as for example Google Authenticator, Microsoft Authenticator and many similar applications. Different Public Key Infrastructure (PKI) solutions can also be used for authentication.

Nowadays, since most users have access to a mobile phone that is connected to the mobile network, user authentication methods based on phone number verification have gained popularity. The objective of phone number verification is to provide an authentication factor based on something that "you have". Using the mobile network infrastructure provides additional benefits, for example a separate, authenticated communication channel. This is an appealing option given the many issues associated with password-based authentication. Phone number verification is often considered a way to complement or even replace traditional password-based methods [2] [3].

Authentication based on phone number verification and the mobile network involves a lot more under the hood than just the phone number. Phone number verification can be based on different methods with significant differences in user and security properties. Three main methods of authentication based on phone number verification are analyzed in this paper:

- SMS-OTP phone number verification.
- Network-based phone number verification.
- SIM-based phone number verification.

1.1 Comparison of Phone Number Verification Methods

While SMS-OTP is widely deployed and adopted, to our knowledge, network-based phone number verification is currently only provided to a limited extent in selected regions. SIM-based phone number verification is, based on our initial analysis, being under consideration for adoption and, so far, not introduced to the market.

A brief comparison of practical properties of the different phone number verification methods is presented in Table 1. It can be especially noted that SIM-based and network-based authentication can be implemented without user interaction.

Regarding security, the conclusion of this paper is that there are significant differences between the methods. First, in Table 2, the susceptibility to different types of attacks is illustrated. Attacks relevant for phone number verification are further described in section 5.

In particular, and as a conclusion of the analysis in this paper, there are significant differences in the security of phone number verification methods against local attacks from malicious applications. A comparison of the security based on the estimated attack complexity is given in Table 3. In this table, SIM-based phone number verification is presented in two versions, to illustrate that the two versions have different security properties in this respect. SIM-based phone number verification v1 is described in section 2.4.1 and SIM-based v2 in section 2.4.2.

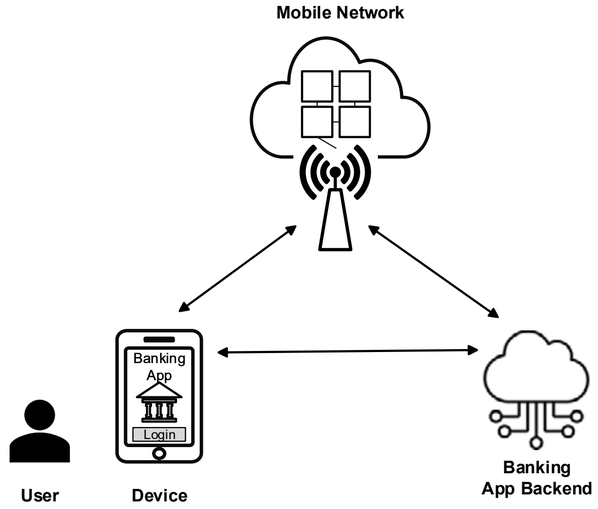


Figure 1. Phone number verification setup overview

Phone Number Verification Method	Maturity	Typical User Interaction for Authentication
SMS-OTP	Widely adopted	SMS input or autofill with or without user action
Network-based	Introduction phase	None or user acceptance
SIM-based	Under consideration	None or user acceptance

Table 1. Comparison of the properties of the different phone number verification methods.

Phone Number Verification Method	Susceptible to SIM-swapping attacks	Susceptible to SMS channel attacks	Susceptible to Phishing attacks	Susceptible to Local attacks by malicious application
SMS-OTP	Yes	Yes	Yes	Yes*
Network-based	Yes	No	No	Yes*
SIM-based	Yes	No	No	Yes*

Table 2. Susceptibility to different types of attacks for the analyzed phone number verification methods.
**While all methods are susceptible to local attacks, the estimated attack complexity differs significantly, see Table 3.*

Phone Number Verification Method	Estimated complexity of local attack by malicious application	Motivation
Network-based	Low	Simple relay of the attacker's traffic through the victim's device can be enough for succesful attack. Required privileges given by default.
SMS-OTP	Intermediate	Attack with malicious application needs access to SMS, which is not granted by default.
SIM-based v1	Intermediate	Relay of the attacker's traffic through the victim's device is not enough, additional actions for authentication needs to be performed on the victim's device.
SIM-based v2	High	Requires the attacker to overcome several security functions, which is considered to imply significant complexity.

Table 3. Estimated attack complexity for local attacks by a malicious application for the analyzed phone number verification schemes. (Low attack complexity can be considered less secure.)

2 Phone Number Verification Methods

In this section the different methods of phone number verification are described, starting with the entities involved in phone number verification.

2.1 Entities of Phone Number Verification

A common factor between the methods is that they use the established mobile network infrastructure as a trusted 3rd party in order to achieve an additional authentication factor.

To illustrate and compare the phone number verification methods in a comprehensive way, the entities of the methods have been simplified to the following:

- **Device:** Mobile phone or network connected device. On the device, there are third party applications, for example financial applications. There are also dedicated applications that handle the connection with the mobile network. In this paper these are referred to as a mobile network operator client.
- **Mobile network operator client:** The mobile network operator client is an application on the device that can have low-level interaction with the SIM-card. The mobile network operator client has a special role in the communication and security of the methods and is therefore sometimes depicted separately from the third-party applications on the device.
- **Mobile network:** The mobile network serves as an abstraction representing distinct entities. This includes the infrastructure for mobile communications, such as the radio interface, base stations, and core network components. Additionally, this abstraction encompasses specific entities responsible for authentication and phone number verification, which vary between different verification methods.
- **Application:** A third-party application that can be downloaded from an application marketplace. The application is installed locally on the mobile device, but is connected to the application backend. Only applications that implement phone number verification are relevant in this paper.
- **Application backend:** The application backend is the server-side component of a software application. It typically runs on cloud services or dedicated servers and is responsible for managing and storing user resources and data. Examples include handling email accounts, processing financial transactions, storing user profiles, and executing business logic. The backend interacts with the frontend (Application) to deliver a seamless experience, ensuring that user requests are processed efficiently and securely.

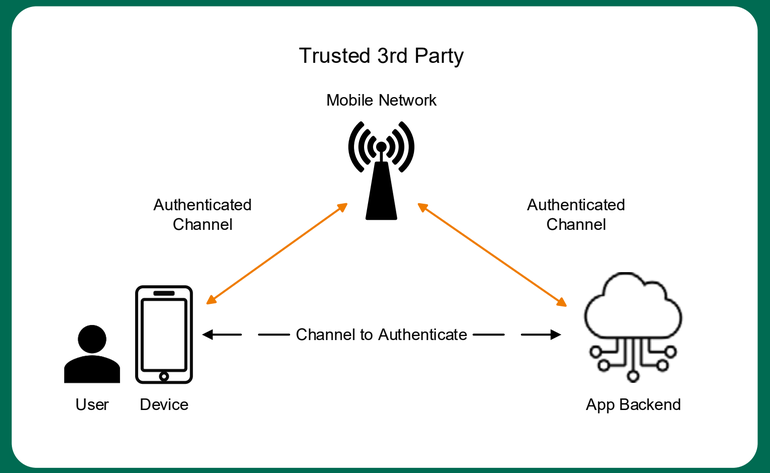


Figure 2. Illustration of the typical entities involved, the main communication channels and the mobile network used as a trusted third party for authentication

Phone Number Verification Methods

2.2 SMS-OTP Phone Number Verification

SMS-OTP phone number verification is based on the verification of a phone number and user intent by using SMS and One-Time Passwords (OTP). In short, this authentication method involves the application backend sending an SMS containing an OTP to the user's phone number via the mobile network (SMS Gateway). The user receives the OTP on their device, enters it into the application, and the application then communicates the OTP back to the application backend, which checks it against the original OTP. This procedure is referred to as SMS-MT (Mobile Terminated).

There is also another version of SMS-OTP called SMS-MO (Mobile Originated). In this method, the application backend provides the OTP to the user, who then sends an SMS containing the OTP back to the mobile network. The network forwards this OTP to the application backend for verification.

The SMS-OTP methods typically have the important property of requiring user interaction. The user must typically either introduce the OTP from the SMS to the application manually, or explicitly give permission for the application to read the SMS.An advantage of this is that the user is aware of the authentication process, which can prevent certain attacks. However, a disadvantage is that the user is exposed to the OTP in plaintext, which can be used in phishing attacks.

One-Time Password is used in several different contexts for authentication, typically based on the methods HOTP (HMAC OTP) and TOTP (Time-based OTP). The security of the phone number verification method SMS-OTP as defined in this paper can, however, not be based on the security analysis of these methods, since there are important differences in how they are used. More information on this can be found in Appendix 4.

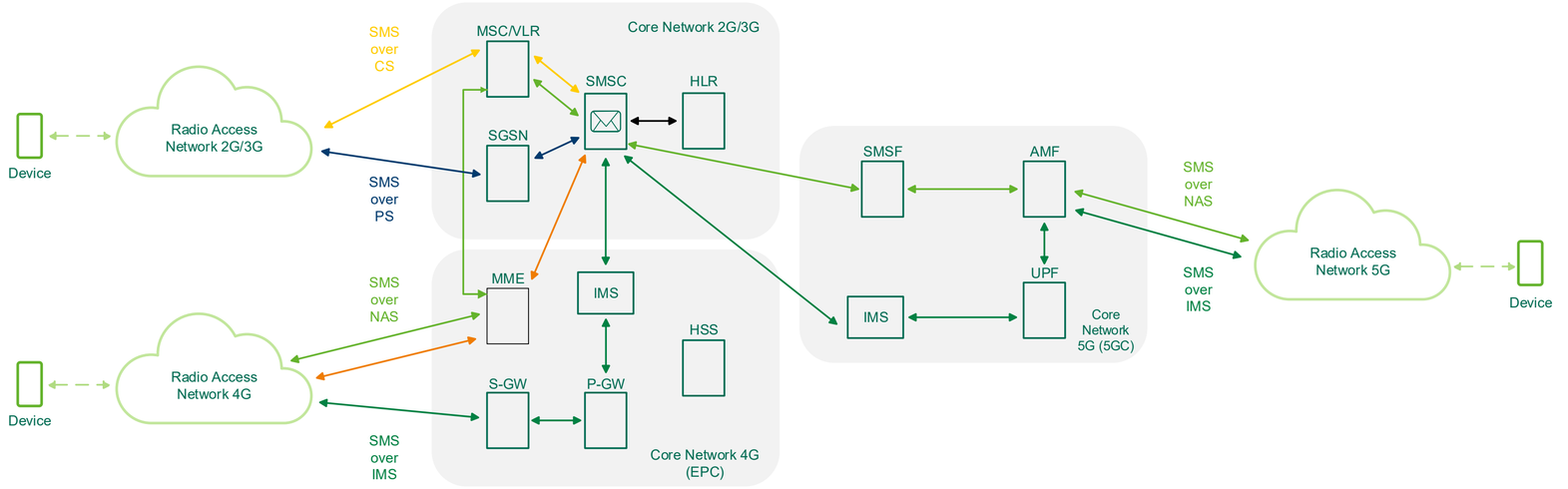


Figure 3. Illustration of the main components for the use of SMS in 2G, 3G, 4G and 5G networks. For more information on mobile networks and how SMS works, see [6].

The SMS-MT procedure for SMS-OTP phone number verification for authentication is described as follows in Figure 2.

1. A user action, for example login to a banking application on the device, triggers an authentication process where the user enters its phone number into an application.
2. The application on the device communicates the phone number to the application backend. Alternatively, the application backend already has the phone number of the user in its database, and there is no need to enter the phone number again. The user action in Step 1 itself triggers the authentication request and the next stage in this case.
3. The application backend generates an OTP, for example a 6-digit code.
4. The application backend contacts the mobile network with the OTP and the phone number provided by the user, or that was given in the database of the application backend.
5. The mobile network sends the SMS with the OTP to the device corresponding to the phone number given by the application backend.
6. The user receives the SMS with the OTP and enters the OTP into the device application.
7. The device application in turn communicates the OTP to the application backend.
8. The application backend then compares the OTP sent to the mobile network with the OTP received from the user. If the OTPs match, the application backend can conclude that the user possesses the device associated with the phone number.
9. The application backend grants access to the device application to the requested resources.

SMS-OTP MO is similar and illustrated in Figure 3.

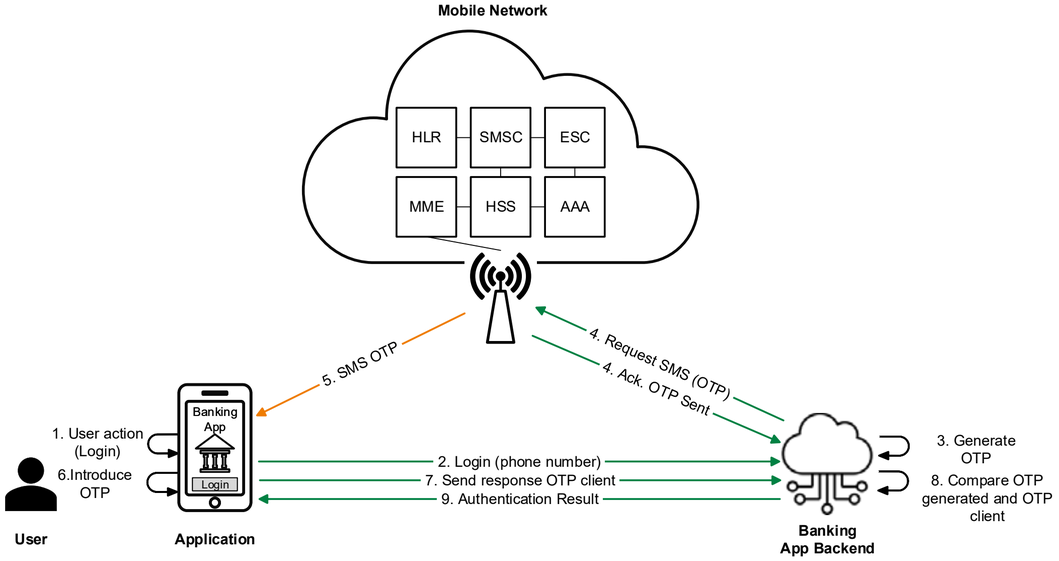


Figure 4. Authentication procedure for SMS-OTP MT phone number verification.

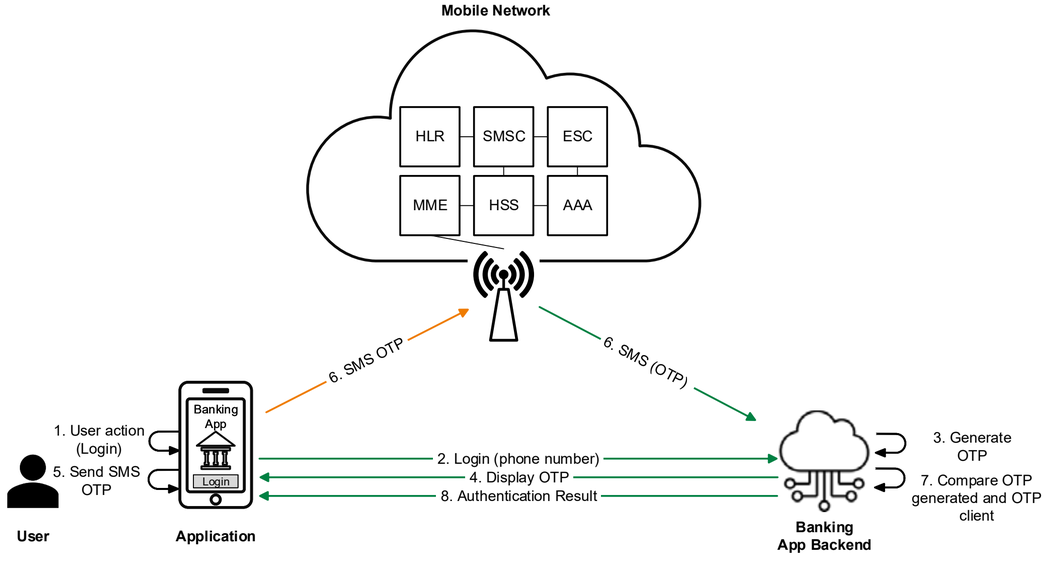


Figure 5. Authentication procedure for SMS-OTP MO phone number verification.

2.3 Network-Based Phone Number Verification

Network-based phone number verification, as defined in this paper, refers to methods where the mobile network operator verifies the device's phone number using device or connectivity information, for example the source IP address, for a specific network request. Network-based phone number verification can contain different solutions and implementations, of varying complexity, including "header enrichment".

A specific feature of certain network-based phone number verification methods is that they are silent, in the sense that no user interaction is required. This means that a user may not even be conscious that the authentication has taken place.

In the context of network-based phone number verification, the objectives can vary. In this paper, focus is on the user authentication using phone number verification. However, phone number verification is also an important tool used by online service providers to ensure the uniqueness of each user account, by connecting a user account to a unique phone number. This helps for example to stop fake accounts, commonly reported to be used for trolling and different kinds of fraud. More on these problems can be found in [4].

An example of the procedure for network-based phone number verification for authentication is as follows.

1. A user action, for example login to a banking application on the device, triggers an authentication process.
2. The user enters its phone number into a device application and the application on the device communicates the phone number to the application backend.

   Alternatively, the application backend already has the phone number of the user in its database, and there is no need to enter the phone number again. The user action itself triggers the authentication request and the next stage.
3. The application backend contacts a network operator to perform phone number verification and provides additional information.
4. The network operator uses the provided information to infer the phone number of the user, for example based on the data communication (e.g. source IP-address) with the device.
5. The network operator communicates the inferred phone number of the user to the application backend.
6. The application backend performs a comparison of the phone number stated by the user and the phone number given by the mobile network. If the phone numbers are identical, the application backend decides to consider the application on the device as authenticated.
7. The application backend grants access to the device application to the requested resources.

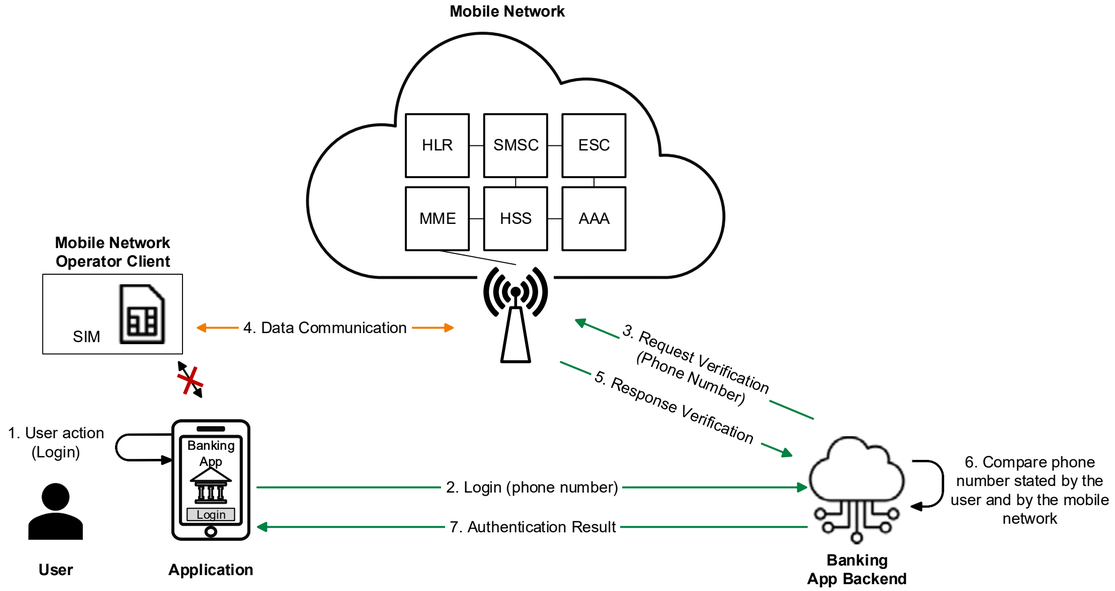


Figure 6. Authentication procedure for one version of network-based phone number verification.

Another, similar, procedure of network-based phone number verification for authentication is using header enrichment and works as follows.

1. A user action, for example login to a banking application on the device, triggers an authentication process.
2. The user enters its phone number into a device application and the application on the device communicates the phone number to the application backend.

   Alternatively, the application backend already has the phone number of the user in its database, and there is no need to enter the phone number again. The user action itself triggers the authentication request and the next stage.
3. The application backend contacts the mobile network to perform phone number verification.
4. The mobile network returns a URL to the application backend, to forward to the application on the device.
5. The application backend relays the URL to the application on the device.
6. The application on the device accesses the URL over cellular connection using http.
7. The network operator communicates the inferred phone number of the user using http with header enrichment to the application backend.
8. The application backend performs a comparison of the phone number stated by the user and the phone number given by the mobile network. If the phone numbers are identical, the application backend decides to consider the application on the device as authenticated.
9. The application backend grants access to the device application to the requested resources.

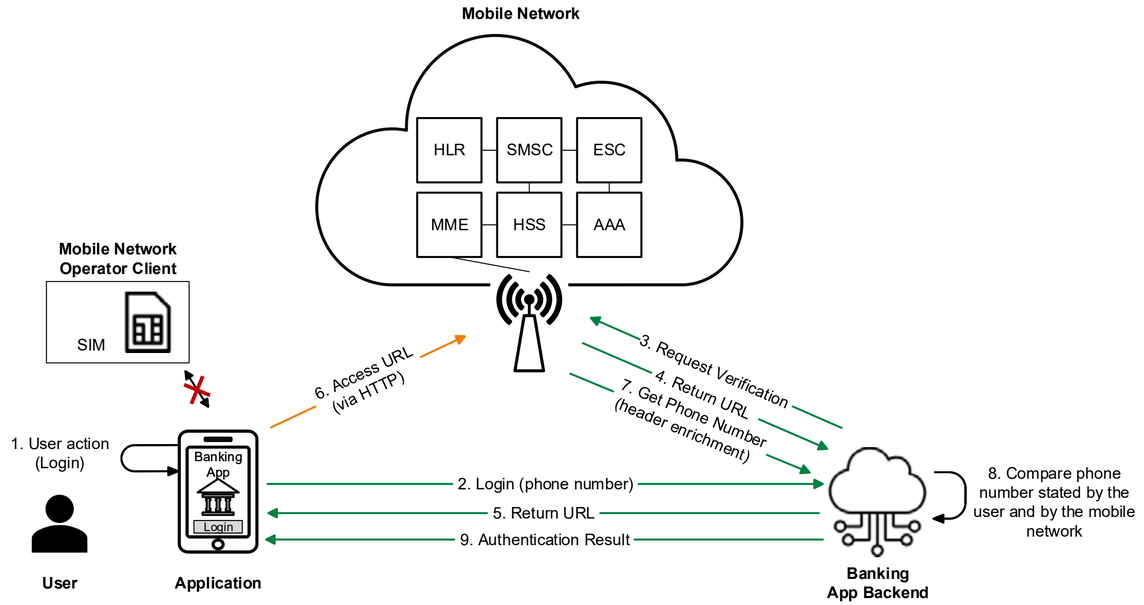


Figure 7. Authentication procedure for a version of network-based phone number verification based on header enrichment.

A possibility of this version of network-based phone number verification is to include a token. This token can be generated in the mobile network after the verification request from the application backend. The token can be stored in the mobile network, and be added to the URL sent to the application on the device, via the application backend. When the application on the device accesses the URL, the token received through the URL access can be compared to the token that was sent, in the mobile network authentication process. This provides a connection between the application on the device, whose phone number is to be verified, and the authentication requests to the application backend and mobile network authentication service. The phone number is then inferred by the source-IP, and then communicated to the application backend.

Another version of the method is to reverse the order in that the mobile application first accesses the mobile network. The mobile network infers the phone number and shares an authentication code with the mobile application. The mobile application then communicates the authentication code to the application backend, that sends the application code and the phone number known to the application backend, to the mobile operator. The mobile operator compares the authentication code generated with the received, and the phone number inferred with the one from the application backend, and returns an authentication response to the application backend.

2.4 SIM-Based Phone Number Verification

SIM-based phone number verification is based on the authentication method from the mobile networks to authenticate mobile phones. A connected mobile phone contains a secret key, securely stored in the SIM-card, and it is shared with the mobile network operator. The usage of this key and identity information of the SIM-card for the authentication is the key property of SIM-based phone number verification.

The mobile network infrastructure has already in place the needed infrastructure, organization and processes for authentication based on symmetric secret keys, for example the key management, including key generation, key storage and the key distribution network of SIM-cards containing the secret keys.

2.4.1 SIM-Based Phone Number Verification Based on Device Authentication

This method for phone number verification is applicable to provide the phone number to, for example, an application backend. The mechanisms are based on the device authentication, but not extended to third party applications, such as a banking application on the device. On the other hand, this method has advantages from a privacy perspective, since the application ID is not necessarily communicated to the mobile network. The description of SIM-based phone number verification in this section is based on [5].

1. A user action, for example a login to a banking application on the device, triggers the authentication process.
2. The user enters its phone number into a device application and the application on the device communicates the phone number to the application backend.

   Alternatively, the application backend already has the phone number of the user in its database, and there is no need to enter the phone number again. The user action itself triggers the authentication request and the next stage.
3. The application requests authentication via the mobile network operator client.
4. This is followed by a SIM-based authentication process, called EAP-AKA, that cryptographically authenticates the device and the mobile network to each other.
5. After this stage, the mobile network operator client requests an AcquireTemporaryToken from the mobile network.
6. The mobile network sends the TemporaryToken to the mobile network operator client.
7. The mobile network operator client forwards the TemporaryToken to the application on the device.
8. The application on the device sends the TemporaryToken to the application backend.
9. After that, the application backend performs a GetPhoneNumber request to the mobile network, including the TemporaryToken
10. The mobile network verifies the TemporaryToken.
11. If the request is authorized, the mobile network respond with the phone number to the application backend. The mobile network operator uses its knowledge of phone numbers corresponding to connected devices to provide this information.
12. The application backend performs a comparison of the phone number provided by the user and the phone number given by the mobile network. If the phone numbers are identical, the application backend decides to consider the application on the device as authenticated.
13. The application backend grants access to the device application to the requested resources.

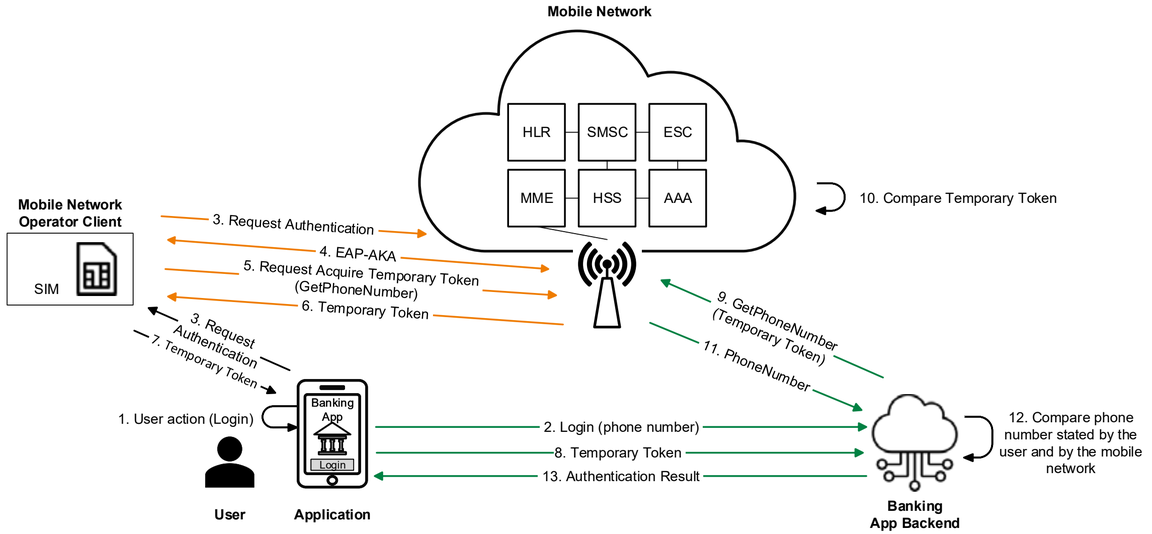


Figure 8. Authentication procedure for SIM-based phone number verification based on device authentication.

2.4.2 SIM-Based Phone Number Verification Extended to Third Party Application

When using SIM-based authentication, the shared key from the SIM-card can be used indirectly by 3rd party applications. While authentication of a mobile device is part of the mobile network operations, further actions are needed in order to extend this authentication to a 3rd party application, for example a banking application.

There are different possibilities of extending authentication to the 3rd party application. In some way, the device application needs to be authenticated to the mobile network. Here, the focus will be on one way to achieve this, which is based on a previous preparation phase, where the secrets needed for the later authentication are shared beforehand.

This description of SIM-based phone number verification for a 3rd party application is based on the description in [5].

Preparation Phase: Provisioning of Secrets

The objective of this phase is to provision the necessary secret tokens to the entities where they are needed for the 3rd party application authentication.

1. A secure channel is created between the mobile network and the application backend.
2. The mobile network sends the OperatorToken to the application backend.
3. The application backend sends the access token to the mobile network and to the device application, using a secure channel assumed between the device application and the application backend.

After this preparation phase, the necessary secrets for the authentication scheme are provided:

- The access token to authenticate the device application to the mobile network.
- The OperatorToken, to finally be given to the device application by the mobile network, to authenticate the device application to the application backend.

Note that instead of steps two and three in the preparation phase, the secure channel can be used to communicate the token as needed in the method described in the next section. Generally, this secure channel is established through an authenticated exchange of public keys, allowing for various methods of creating a secure channel, including the use of encrypted and signed tokens via the application on the device.

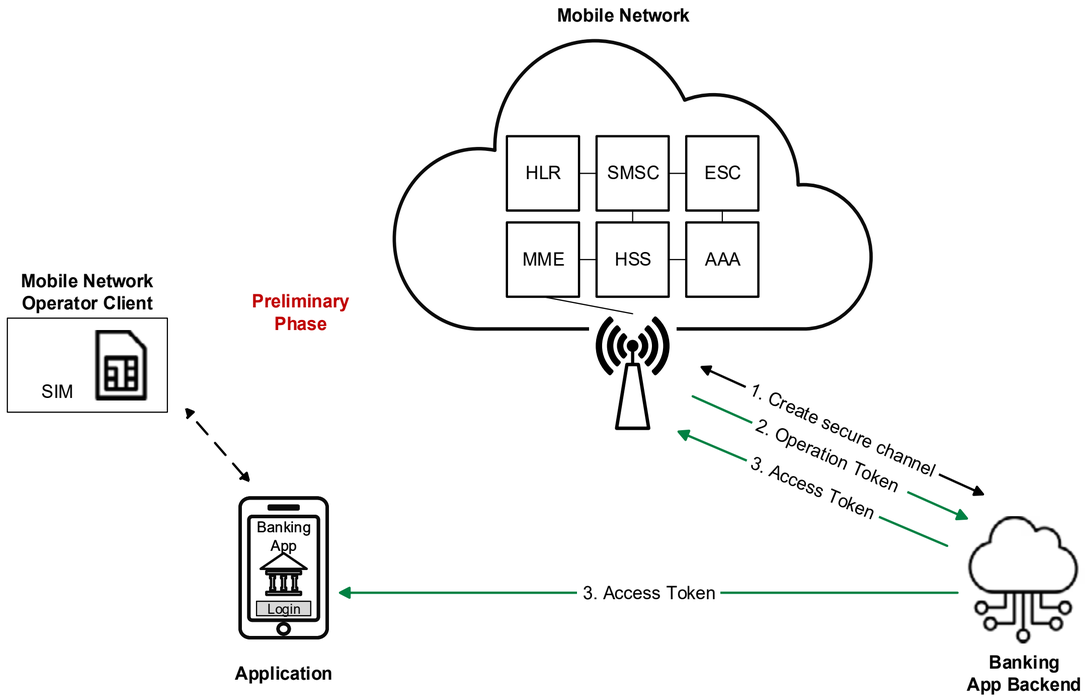


Figure 9. Preparation phase for SIM-based phone number verification.

Description of SIM-Based Phone Number Verification Extended to a Third-Party Application

1. A user action, for example a login to a banking application on the device, triggers the authentication process.
2. In the first part of the authentication process, the bank's application requests authentication via the mobile network operator client.
3. Then the mobile network operator client on the device initiates a SIM-based device authentication. This process starts with the mobile network operator client sending an authentication request to the mobile network.
4. This is followed by a SIM-based authentication process called EAP-AKA that cryptographically authenticates the device and the mobile network to each other. This authentication is the same as the authentication process when connecting to the network and is based on the secret shared keys of the SIM-card and involves the knowledge of the user's phone number.
5. After this stage, the device and the mobile network has established a secure channel.
6. The application uses this channel to send the access token to the mobile network. This is needed for the process of authenticating not only the device, but also the application behind the authentication request to start with.
7. An entity in the mobile network verifies the access token sent from the device with the stored access token in the mobile network. If the access tokens are identical, the mobile network concludes that the application is authenticated.
8. After the application authentication to the mobile network entity, the OperatorToken (also secret) is sent to the device application, from the mobile network entity. This secret token was pre-shared between the mobile network and the application backend.
9. The device application then finally sends the obtained OperatorToken to the application backend.
10. In the final stage, the application backend compares the OperatorToken obtained from the device application with the pre-shared OperatorToken from the mobile network from the previous provisioning phase. After the OperatorToken is received by the application backend, the application backend can additionally request a validation of the OperatorToken with the mobile network, or request network information about the device used, including the phone number, and perform additional comparisons with reference data.
11. If the secret tokens match, the application backend confirms a successful SIM-based phone number authentication and grants the user access to the requested resources. The OperatorToken serves as proof to the application backend that the device application is running on an authenticated device with a verified phone number, and that the application has been authenticated by the mobile network.

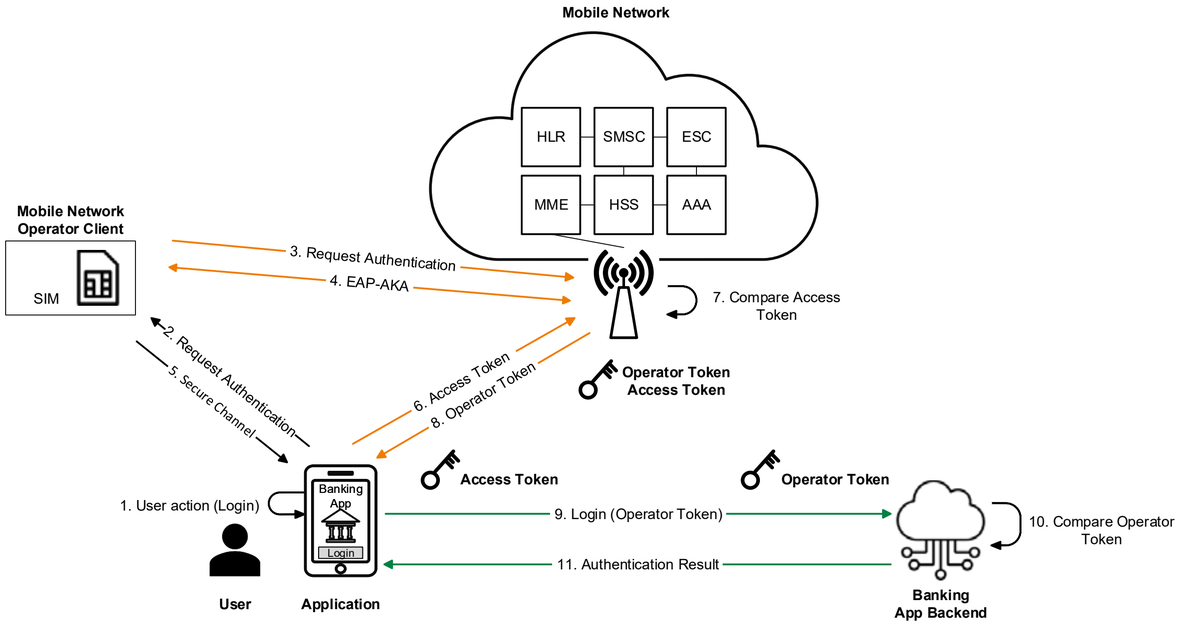


Figure 10. Authentication procedure for SIM-based phone number verification extended to third party application.

The use of access token is optional and can be left out under certain conditions. The application then uses a token obtained from the EAP-AKA process, AuthToken, to obtain the OperatorToken from the mobile network.

3 Required Security Properties - Specific for the Methods

In this section a number of important security properties that are considered necessary for secure implementations of the authentication methods based on phone number verification are identified and described. The objective of this analysis is to capture and illustrate some main underlying security properties and dependencies. This will later be used for the threat analysis.

A number of important security properties are the same, or similar, for all the phone number verification methods. These properties are presented in Appendix 1. This means that the described properties of for example SIM-based phone number verification consists of the general properties together with the specific properties for SIM-based phone number verification.

The security properties that should hold for all phone number verification methods are illustrated in the figure, and explained further in Appendix 1. It is important to note that phone number verification security is dependent on a number of separate entities. This makes security complex and based on the trust of each component to guarantee the overall security.

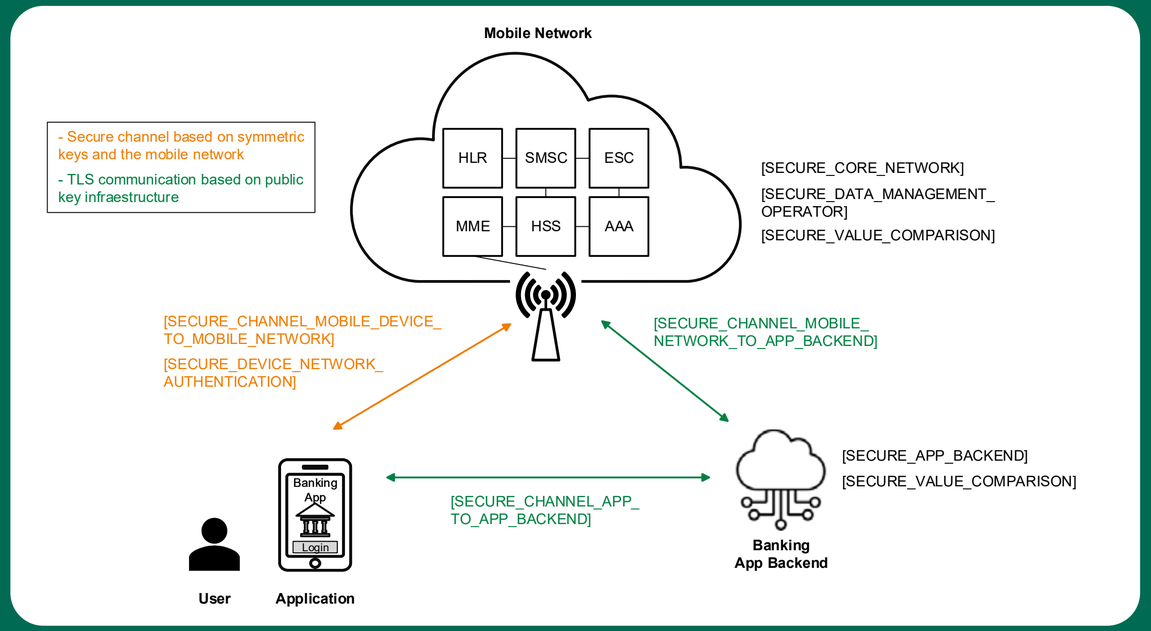


Figure 11. Security properties applicable to all studied methods.

3.1 SMS-OTP Phone Number Verification

The identified and described security properties that are specifically related to SMS-OTP phone number verification are presented in this section. Note that the general properties in Appendix 1 also apply.

Security Property Identifier	Definition of Required Security Property	Security Solution
[SECURE_SMS_TRANSMISSION]	The mobile network transmission scheme for SMS is secure.	SMS has no inherent security, but there are other network security functions that partially protect the security of SMS. See [6] for more information on SMS security.
[OTP_SECURE_GENERATION]	The OTP generation is secure.	There are a number of properties that need to be fulfilled for the security of an OTP-solution. These assumptions involve using a secure cryptographic algorithm for the OTP generation, high entropy input and the OTP must have a sufficiently short validity time.
[SMS_OTP_ACCESS]	Only the intended, authenticated application, or user, has access to the SMS-OTP on the device.	Functionality that restricts access to SMS for applications, and only grants access after explicit user consent or notification.

Table 4. Required security properties for SMS-OTP phone number verification.

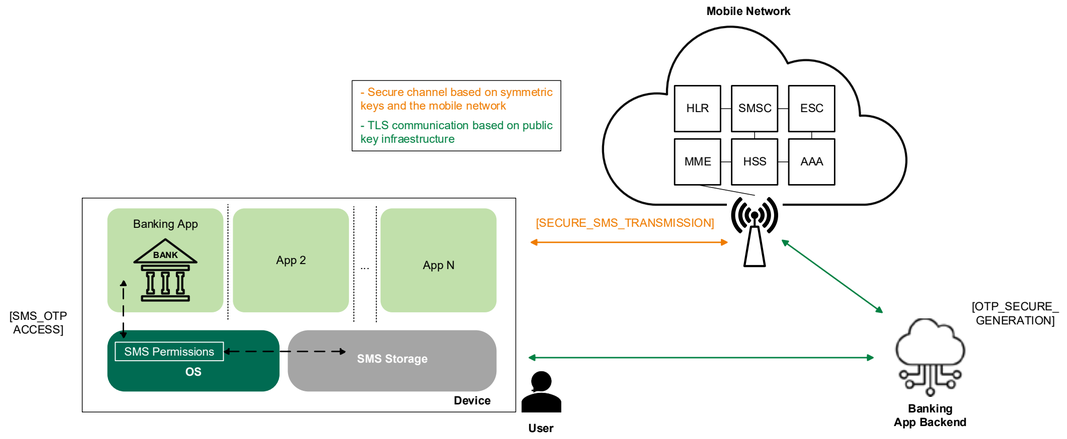


Figure 12. Security properties for SMS-OTP phone number verification.

3.2 Network-based Phone Number Verification

Apart from the security properties that apply to all phone number verification methods, a specific security property for network-based phone number verification is identified and presented below.

Security Property Identifier	Definition of Required Security Property	Security Solution
[OPERATOR_PHONE_ NUMBER_VERIFICATION]	The method that the mobile network operator uses to obtain the phone number and to verify the device, on which the application is running, should be secure.	A common feature of network-based phone number verification is that the mobile network infers the phone number based on the source IP address of the device making a specific network request. Note that the source IP of a request alone does not constitute a secure authentication method.

Table 5. Required security properties for network-based phone number verification.

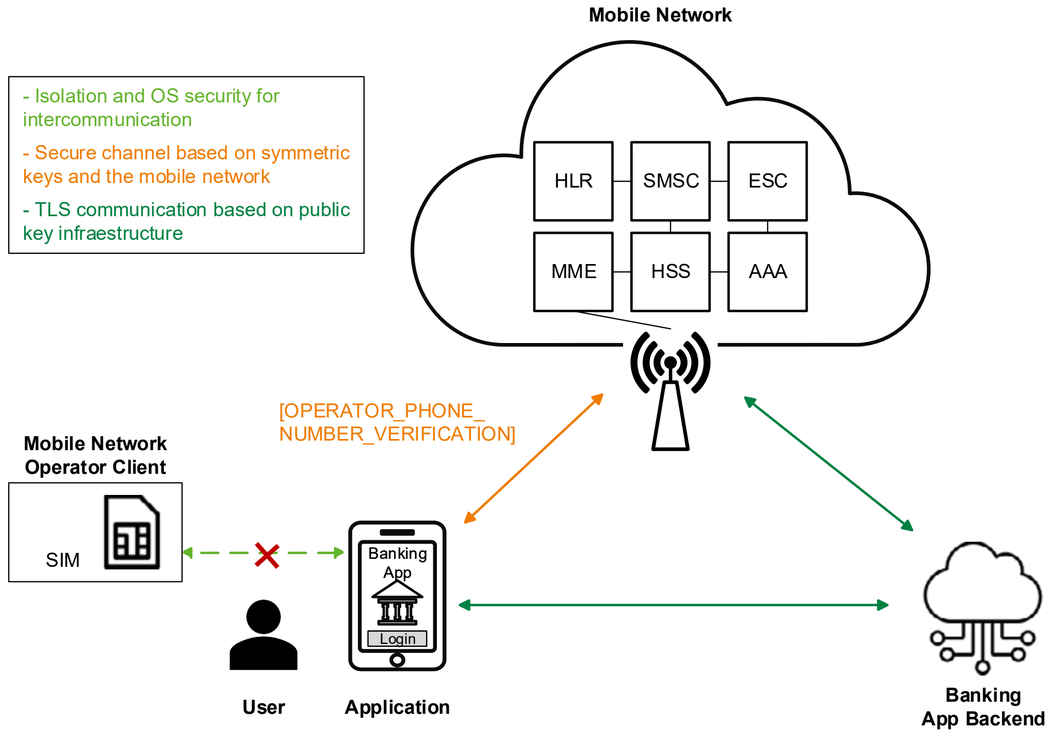


Figure 13. Security properties for network-based phone number verification.

3.3 SIM-based Phone Number Verification

In this section the security properties that are specifically related to SIM-based phone number verification are presented. Note that the general properties in Appendix 1 also apply.

Security Property Identifier	Definition of Required Security Property	Security Solution
[TRUSTED_CLIENT_SIM_ACCESS]	There is a trusted mobile operator client, or application, on the device, that has the privileges to communicate with the SIM-card and the mobile network. Additionally, there are no malicious applications with these privileges.	Approved and signed mobile network client as part of the operating system.
[APP_IDENTITY_TO_TRUSTED_CLIENT]	The mobile network client on the device securely verifies the unique identity of an application.	An application-unique ID and cryptographic verification of binary, ID and signature of the application.
[APP_IDENTITY_AUTH_PROCESS]	The verified identity (e.g. application-unique ID) of the application is part of the authorization process as a whole, and the application identifier is protected.	The application ID is included in the authentication and authorization process in a way that protects the integrity of the application ID, and the corresponding authorization properties, in all places where transmitted and stored. This includes a preparation phase between the application backend and the mobile network.
[SECURE_CHANNEL_APP_TO_TRUSTED_CLIENT]	The channel between the mobile network client on the device and the authenticated application on the device is secure.	Isolation mechanisms implemented by the operating system between applications and core functions are used to achieve this.

Table 6 (1/2). Required security properties for SIM-based phone number verification.

Security Property Identifier	Definition of Required Security Property	Security Solution
[TOKEN_APP_GENERATION]	Potential tokens used are generated securely.	The tokens are generated using sufficient entropy and using the appropriate metadata such as access privileges and time validity.
[TOKEN_APP_PROTECTION]	On the device, potential tokens used, such as the access token and the OperatorToken, are protected.	Usually mobile phones provide hardware backed secure storage mechanisms. This is without considering that SIMs include their own secure storage facilities. Mobile phones support secure storage mechanisms such as Hardware Security Modules (HSMs) and Secure Elements (SEs) for tamper-resistant key storage, Trusted Execution Environment (TEE) for isolated secure processing, and secure storage APIs like Android's Keystore and iOS's Keychain for encrypting sensitive data. They also use file-based encryption (FBE) and full disk encryption (FDE) to protect data at rest, and biometric authentication methods (fingerprint and facial recognition) to enhance security. These technologies ensure that sensitive information is securely stored and protected from unauthorized access and tampering.

Table 6 (2/2). Required security properties for SIM-based phone number verification.

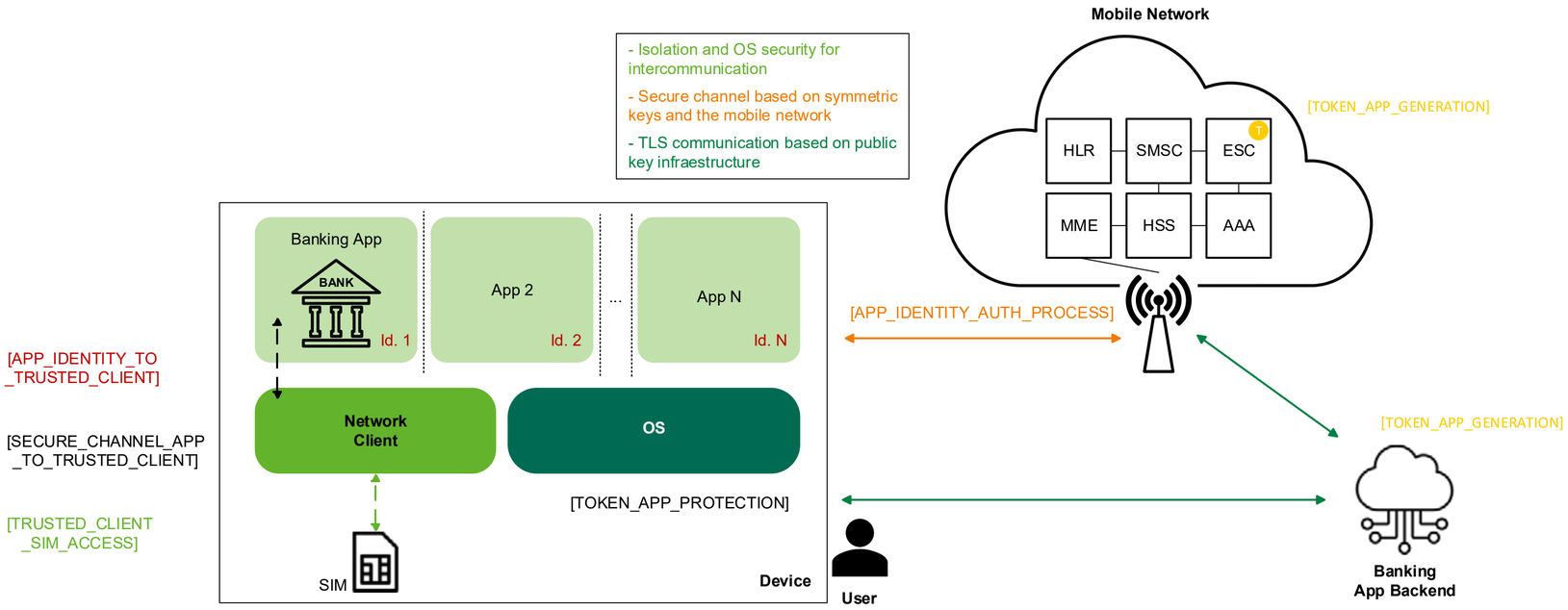


Figure 14. Security properties for SIM-based phone number verification.

4 Threat Analysis

Based on the previous analysis of the security properties for the evaluated phone number verification methods, a corresponding threat analysis is performed in this section. A number of different relevant threat scenarios were identified and described. The threat scenarios do not apply one-to-one to the security properties, instead similar threat scenarios apply to several security properties. This is illustrated and explained for each threat scenario and later summarized in a table.

The threat scenarios in this section do not include every threat, but illustrate the most relevant threats to phone number verification. The threat scenarios are to be understood in the context of absence of other authentication factors, or other security mechanisms. Additional security measures such as for example monitoring of network and data activity patterns, additional secure channels etc. may mitigate some of the threats, or at least increase the complexity and cost of performing successful attacks. In fact, the threat scenarios serve to illustrate the importance of using security functions and security measures across all the entities involved in the phone number verification methods.

Some of the threat scenarios apply to all phone number verification methods, while some attacks are specific. The applicability of the threat scenarios is presented in Table 7.

#	Threat Scenario	Phone Number Verification Method
1	Attacker with SMS access on the device.	SMS-OTP
2	Phishing attack to obtain SMS-OTP from user.	SMS-OTP
3	Attacker with network access on the device.	Network-based
4	SIM-based phone number verification spoofing via malicious application.	SIM-based
5	Attacks on authentication data.	All
6	SIM-swapping attacks.	All
7	Attacks on the secure channels between the mobile network, the application backend and the application.	All
8	Attacks against critical decisions functions.	All
9	Attacks against the secure channel between the device and the mobile network.	All

Table 7. Applicability of the threat scenarios for phone number verification methods.

In a later section, attacks based on the threat scenarios will be presented and demonstrated with proofs-of-concept for a selection of the attacks.

The threat scenarios that apply to all methods, threat scenario 5 to 9, are found in Appendix 2.

4.1 Threat Scenario 1: Attacker with SMS Access on the Device

An efficient way to attack phone number verification based on SMS-OTP is to use a malicious application on the victim´s device. Currently, 3rd party applications are not given privileges by the operating system to read or write SMS, unless specifically approved by the user (this holds for Android and iOS). However, a user can be deceived to grant SMS privileges to the malicious application. The malicious application may provide some functionality desired by the user, while secretly serving as a platform for attacks.

An attacker with an application that has the privileges to read SMS, can retrieve the SMS-OTP and send it to another device that authenticates as the victim's phone, getting access to the victim´s resources.

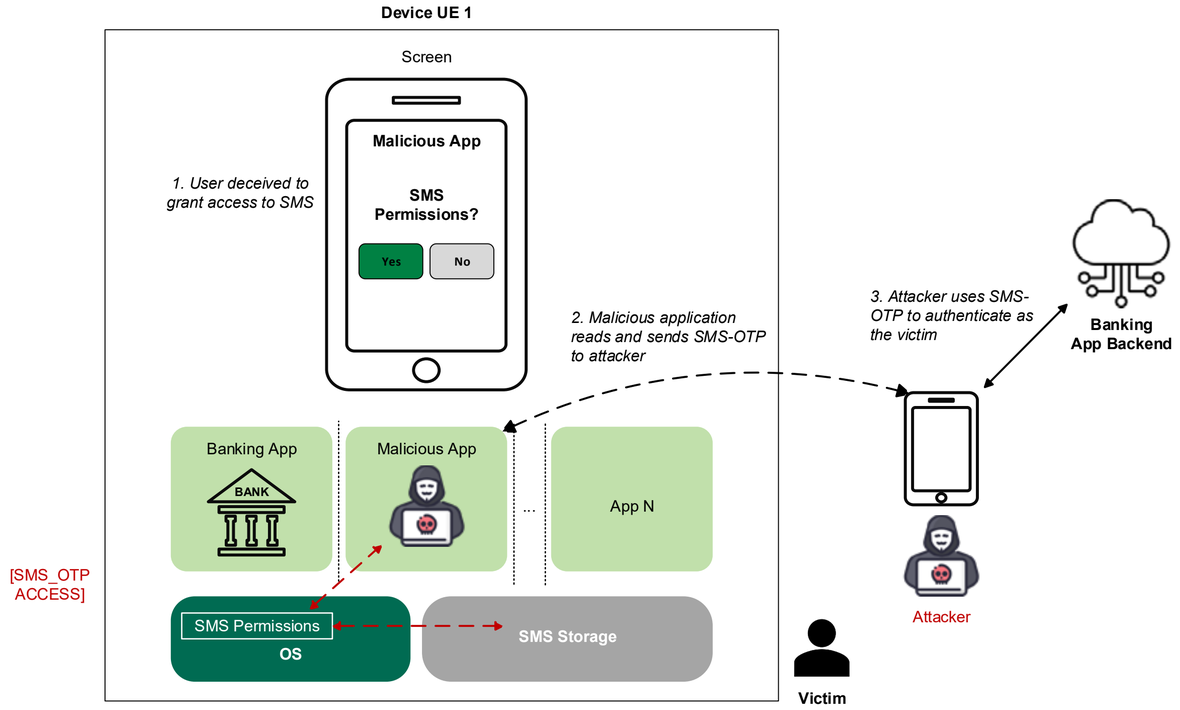


Figure 15. The attacker obtains the SMS via a malicious application installed on the victim's mobile phone.

4.2 Threat Scenario 2: Phishing Attack to Obtain SMS-OTP from User

Another relevant threat scenario against SMS-OTP phone number verification is phishing attacks, in which an attacker deceives the victim to provide the OTP-value. Attackers have demonstrated to be very skilled and successful in performing these types of attacks. A typical attack procedure consists of the attacker claiming to represent the bank, or some other service, and that they call to support and protect the victim against an ongoing attack. In reality, the attacker is deceiving the victim and performing the attack using the victim´s input.

The attacker prepares an authentication process to the victim´s resources in parallel. When the SMS-OTP is obtained, it is used by the attackers to authenticate and access the victim´s resources, in the same way as a standard authentication.

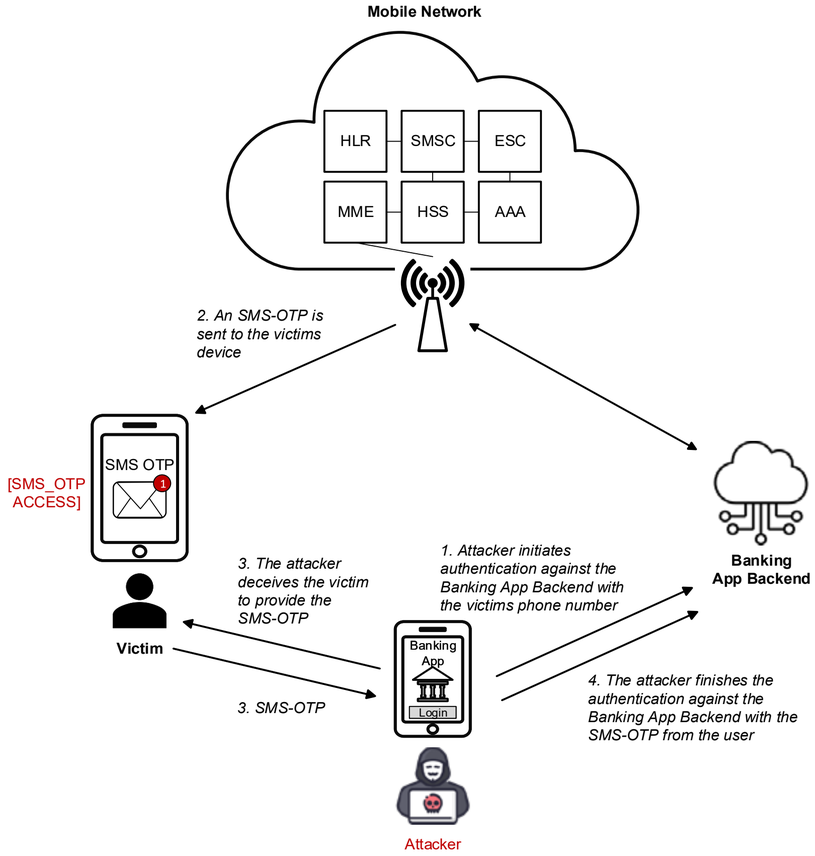


Figure 16. Phishing attack to obtain SMS-OTP from victim.

4.3 Threat Scenario 3: Attacker with Network Access on the Device

In this threat scenario against network-based phone number verification, the attacker uses the victim‘s phone to relay the communication from the attacker´s device, where the attacker performs an authentication process, using the legitimate application, but providing the victim‘s phone number. The network-based phone number verification will confirm that the traffic is from the victim‘s phone, and the victim is unaware of the attack.

The basis for this threat scenario is that the network-based phone number verification only infers the phone number based on the IP-address of the network access, to which the attacker has access, and there is no mechanism for the mobile network to detect this for this method.

Most applications require network access in order to be useful, for example to connect to the application backend and therefore network access is generally granted by default to 3rd party applications. This makes this threat scenario very relevant.

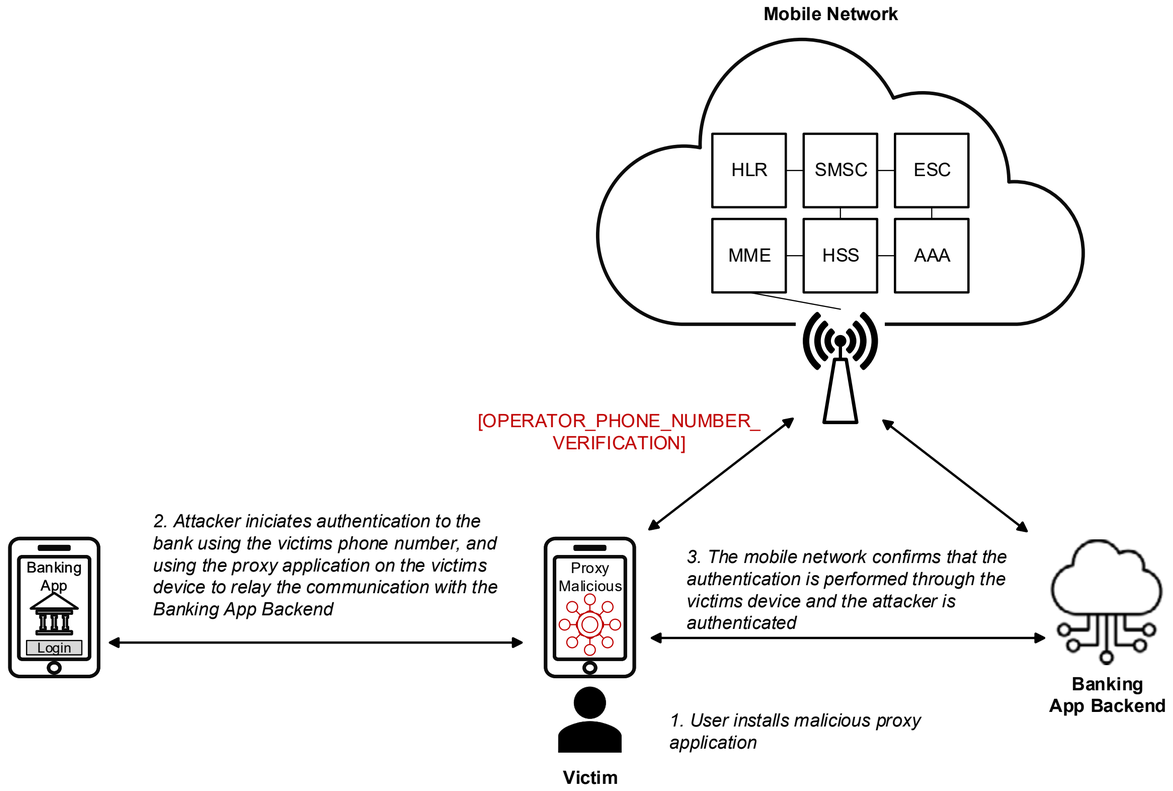


Figure 17. An attacker logs into the banking application through the proxy to impersonate the victim.

4.4 Threat Scenario 4: SIM-Based Phone Number Verification Spoofing via Malicious Application

Attacks against SIM-based phone number verification by an attacker using a malicious application on the mobile device can be considered complex. This is due to the need to bypass multiple security measures.

In the case of using SIM-based phone number verification based on device authentication, a local attacker could pretend to be a legitimate application and try to perform the phone number verification. This could be possible if there is no device-internal authentication of the application performing a GetPhoneNumber request.

A malicious application can relay the data traffic through the victim´s phone, but an additional request of GetPhoneNumber needs to be performed to the mobile network operator client locally at the victim´s device for a successful attack. Hence it seems that the attack complexity is higher than for network-based phone number verification. However, a malicious application that can both relay the data traffic, and perform the request to the mobile network operator client, could perform an attack.

In the other case of using SIM-based phone number verification extended to third party applications, using the legitimate application “as is” in the attacker´s phone and using a malicious application in the victim´s phone to relay traffic is also not sufficient. Simply extracting assets from the victim‘s phone and inputting them into the attacker‘s legitimate banking app is also not sufficient.

Instead, attacks on the victim´s device are required to achieve necessary assets and privileges for the authentication process, and additionally, it seems that the attacker needs to implement necessary communication processes and logic of the legitimate application to perform the attack. Among the possible threat scenarios to achieve assets needed by the attacker a few possibilities are outlined.

One threat scenario involves an attacker using a malicious application with privileges to access the SIM card. The attacker could then execute attacks on the SIM-based phone number verification scheme as follows: the malicious application spoofs the SIM card as the legitimate application. If an access token is required, the attacker would also need to obtain this token from the legitimate application to have all the necessary assets for successful SIM-based phone number verification.

We note, however, that the privilege of access to the SIM-card is not granted to 3rd party applications in most relevant OS used nowadays, independently of user actions. Because of this, it is necessary to exploit some privilege escalation vulnerability in the device to obtain these privileges. In order to obtain the access token from the authentic application, an additional attack against the device security is also necessary, in case an access token is used.

Another threat scenario involves an attacker, using a malicious application, spoofing the mobile network client to appear as the legitimate banking application. This allows the attacker to execute a similar attack as with SIM card access. However, if an access token is required, the attacker must also perform an additional attack to obtain the access token from the legitimate application.

Yet another threat scenario is that an attacker gains access to the necessary authentication tokens while stored or handled by the device, for example, through a malicious application. One way of achieving this would be to gain access to the communication between the legitimate application and the mobile network client, or access to the memory where the assets are stored. With the necessary tokens, the attacker can perform an attack against SIM-based phone number verification.

Note, however, that also this threat scenario would need additional exploitation of vulnerabilities in order to have access to these assets, since most modern operating system implements isolation mechanisms at different levels including applications.

In summary, the identified threat scenarios for SIM-based phone number verification requires the attacker to overcome several security functions and are considered to have a significantly higher complexity level than the SMS-OTP and network-based phone number verification.

4.5 Summary of Threat Scenarios and Security Properties

The table below maps the threat scenarios to the identified security properties. It includes both the specific security properties and threat scenarios previously presented, as well as those applicable to all methods, which are detailed in the Appendix 2 for completeness.

#	Threat Scenario	Security Properties
1	Attacker with SMS access on the device.	[SMS_OTP_ACCESS]
2	Phishing attack to obtain SMS-OTP from user.	[SMS_OTP_ACCESS]
3	Attacker with network access on the device.	[OPERATOR_PHONE_NUMBER_VERIFICATION]
4	Attacker on the device against SIM-based phone number verification.	[SECURE_CHANNEL_APP_TO_TRUSTED_CLIENT] [APP_IDENTITY_TO_TRUSTED_CLIENT] [TRUSTED_CLIENT_SIM_ACCESS] [TOKEN_APP_PROTECTION]
5	Attacks on authentication data.	[SECURE_DEVICE_NETWORK_AUTHENTICATION] [SECURE_DATA_MANAGEMENT_OPERATOR] [SECURE_CORE_NETWORK] [SECURE_APP_BACKEND] [APP_IDENTITY_AUTH_PROCESS] [SECURE_SMS_TRANSMISSION]
6	SIM-swapping attacks.	[SECURE_DEVICE_NETWORK_AUTHENTICATION] [SECURE_DATA_MANAGEMENT_OPERATOR]
7	Attacks on the secure channels between mobile network, the application backend and the application.	[SECURE_CHANNEL_MOBILE_NETWORK_TO_APP_BACKEND] [SECURE_CHANNEL_APP_TO_APP_BACKEND]
8	Attacks against critical decision functions.	[SECURE_VALUE_COMPARISON] [SECURE_APP_BACKEND] [SECURE_CORE_NETWORK] [OTP_SECURE_GENERATION] [TOKEN_APP_GENERATION]
9	Attacks against secure channel between the device and the mobile network.	[SECURE_SMS_TRANSMISSION] [SECURE_CHANNEL_MOBILE_DEVICE_TO_MOBILE_NETWORK] [SECURE_DEVICE_NETWORK_AUTHENTICATION]

Table 8. Required security properties corresponding to the threat scenarios.

5 Publicly Reported Attacks

Based on the threat scenarios, a number of different attacks that can be applied in the threat scenarios for phone number verification have been identified. The relevance of these attacks to the threat scenarios and the security properties can be seen in Table 9. In sections 5.1 to 5.8 the identified attacks are described and publicly reported attacks are referenced where relevant and identified.

#	Threat Scenario	Attacks against phone number verification using
1	Attacker with SMS access on the device.	Local attacks
2	Phishing attack to obtain SMS-OTP from user.	Phising attacks
3	Attacker with network access on the device.	Local attacks
4	SIM-based phone number verification spoofing via malicious application.	Local attacks
5	Attacks on authentication data.	Data interception and modification Network attacks Man-in-the-middle attacks
6	SIM-swapping attacks.	SIM-swapping attacks
7	Attacks on the secure channels between the mobile network, the application backend and the application.	Network attacks Man-in-the-middle attacks
8	Attacks against critical decisions functions.	Brute-force attacks Network attacks Replay attacks
9	Attacks agains the secure channel between the device and the mobile network.	Data interception and modification

Table 9. Attacks corresponding to the threat scenarios.

5.1 Local Attacks

Local attacks require direct access to the mobile device and typically exploit vulnerabilities in downloaded apps or use malicious applications that often provide some legitimate functionality. In [7], data breaches via malicious apps was ranked as the second most common cyberattack on mobile devices in 2021. This issue is partly due to users commonly accepting app permissions without fully understanding the potential risks.

In [8], a systematic analysis of the security of local attackers against SMS-OTP was performed. In the paper, it is noted that since the use of SMS-OTP has increased, both iOS and Android have introduced mechanisms to prevent 3rd party applications from having access to SMS messages.

In the same reference, several significant findings were uncovered, including vulnerabilities in the handling of SMS-OTP on Android, which have since been fixed. One of these findings was related to the identification of applications, where a truncated hash value was previously used as identifier of applications to obtain access to SMS. Since the hash value was truncated to only 66 bits, practical attacks to produce an application that has the same identifier could have been possible, although computationally expensive. These attacks could have been used in a spoofing attack against the operating system to obtain access to SMS.

5.2 Phishing Attacks

In phishing attacks against SMS-OTP, the primary goal of the attacker is to obtain the OTP code from the victim. A classical procedure of a directed phishing attack is to deceive the victim through a call. However, another phishing attack is to make the victim install a malicious application and to grant the privilege to access SMS.

In [8], an interesting study to evaluate the likelihood that a user is lured into granting SMS permission to a malicious application was performed. In their experiments with users, 45% to 71% percent of the users granted access to the malicious application. This number was not affected by mechanisms such as “one-tap” introduction or autofill, and was also consistently high across operating systems.

5.3 SIM-swapping Attacks

In SIM-swapping attacks, attackers take advantage of the security of mobile operators to transfer a user‘s phone number to a SIM-card in their possession. This can be accomplished through various methods, such as contacting the victim‘s mobile operator and requesting the change. For the social engineering aspect of this attack to be successful, the attacker typically needs specific information about the victim, such as their phone number, ID card number and other personal details.

Another method is attempting to bribe mobile operator employees to hijack SIM cards. This is reported in [9].

As an example, the operator T-Mobile has suffered several SIM-swapping attacks in recent years. In February 2021 they suffered an attack through an internal application that targeted 500 customers [10]. The FBI said that, in the same year, it received 1611 complaints about SIM-swapping, with claimed losses of $68 million [11]. Another example is given in [12].

The effect of SIM-swapping attacks is that the victims´ SIM-card is deactivated and the attacker receives all calls and SMSs, including the use as authentication factor, to obtain password reset codes, receive banking information etc.

5.4 Brute-force Attacks

Brute-force attacks are attacks based on exhaustive search over all possible combinations, for example of a code or SMS-OTP. This can be an efficient attack if the number of possibilities is sufficiently low, and if the test of possibilities can be performed in a sufficiently efficient manner. A secure implementation detects brute-force attacks by for example limiting the number of attempts.

Brute-force attacks have been studied in [13]. In their study, they found out that in many cases SMS-OTPs were incorrectly generated, either by missing entropy or insufficient OTP length. Furthermore, the comparison functions were sometimes vulnerable to brute-force attacks due to for example too many retry attempts. Similar results were found in [14], where SMS-OTPs were notably the same SMS-OTP could be used for authentication, even from a different session.

5.5 Replay Attacks

A replay attack is a type of attack where the attacker captures a message or value and later replays it. Depending on the scenario, replay attacks can work even with encrypted data, if the proper security mechanisms are not implemented. For authentication, a replay attack can consist of the attacker resending a code, for example an OTP, that has already been used, to grant access.

Replay attacks have been studied in [13]. In this study, they found that of the 544 Android applications studied, 106 accepted a repeated OTP, which made them vulnerable to replay attacks.

5.6 Man-in-the-middle Attacks

In a man-in-the-middle attack the attacker secretly intercepts and forwards the communication between two parties. In the setting of phone number verification this can be performed on different channels, using different methods. A requirement for a successful man-in-the-middle attack is that the channel is not already authenticated and secure. This is the case in for example unprotected Wi-Fi networks. Another option is that the attacker impersonates the legitimate party before the secure channel is setup, and then relays the traffic.

For the radio channel between a mobile device and the mobile network, man-in-the-middle attacks can be performed on 2G networks, for example using so-called StingRays, that simulates a base station. This is further described in [6].

For the channels based on communication between a user and an application backend, man-in-the-middle attacks can be performed for example by providing a similar domain-name, for example goodle.com, instead of google.com, and making the user believe that the correct domain is accessed. The attacker can this way relay the user credentials to the authentic web application backend, obtaining access and circumventing some authentication methods.

5.7 Data Interception and Modification

In the context of SMS interception, it involves an attacker gaining unauthorized access to SMS messages transmitted to or from a victim. This attack can compromise the security of various services that rely on SMS-OTP for two-factor authentication (2FA).

A real case of such an attack occurred in 2017, when hackers intercepted second-factor authentication credentials to authenticate transactions from the bank accounts of a German bank to accounts controlled by the attackers. This exploitation allowed the criminals to fraudulently divert funds, exposing the vulnerabilities in SMS-based authentication systems. In this case, hackers exploited vulnerabilities in the SS7 (Signaling System 7) telecom protocol to intercept second-factor authentication (2FA) codes sent via SMS [15].

In July 2024, security researchers from CCC (Chaos Computer Club) reported [16] that they had obtained more than 200 million SMS messages for authentication from more than 200 companies from the SMS provider IdentifyMobile. The provider distributed the OTP codes in real time over the internet and the CCC team was able to intercept them.

5.8 Network Attacks

Network attacks consist of a large number of different attack techniques to gain remote access to information resources. Network attacks are commonly present across industries and computer infrastructures in our society. When directed to mobile operators, in the perspective of phone number verification, there is an additional risk of compromising the authentication information of customers. The same holds for application backends. Attacks on operators are not uncommon, although the consequences of the attacks depend on the attackers´ objective.

In [17], it is reported that staff from operators working with network engineering and IT infrastructure management were targeted and their credentials leaked on the Dark Web. An attack against Spanish telecom provider Orange in 2024 is reported in [18]. It is reported in [19] also that all the big 5G operators in the US, Verizon, T-Mobile, AT&T and Dish Network, were involved in some kind of security incident in 2023.

Another example is a destructive attack from December 2023, when Kyivstar´s mobile and data services went down. This caused all of its subscribers, more than 25 million, to be left without Internet. This attack consisted of destroying more than 10 000 computers, 4 000 servers and all cloud storage and backup systems [20].

Other types of attacks are those that affect backend applications, such as password leaks through SQL injection attacks, cross-site scripting, remote code execution, etc. According to Verizon‘s annual data breach investigations report [21], these types of attacks have doubled since 2022.

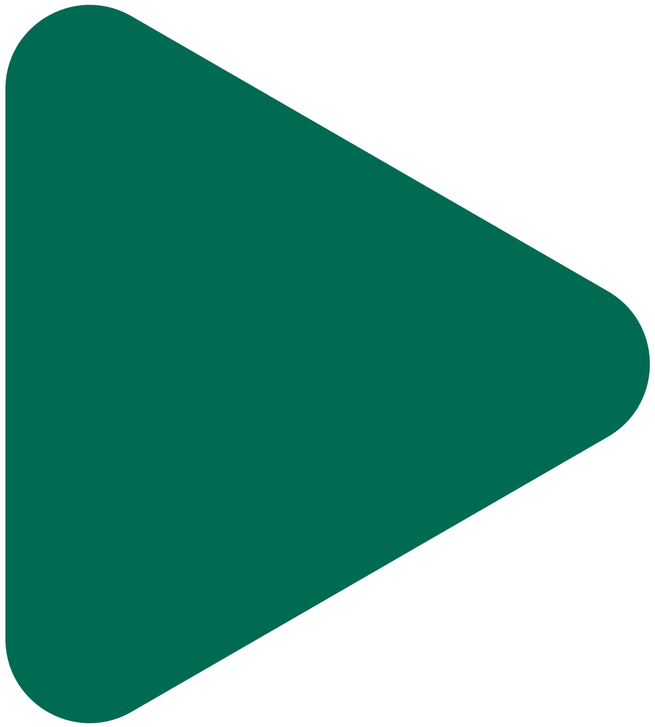
6 Proofs of Concept

In this section, we focus on proofs of concept that highlight specific and distinct attacks for different phone number verification methods. The proofs of concepts described are the following:

- Network-based attack, which follows the threat scenario “Attacker with network access on mobile device”.
- SMS-OTP attack, which follows the threat scenario “Attacker with SMS access on device”.
- SMS-OTP phishing attack, which follows the threat scenario “Phishing attack to obtain SMS-OTP from user”.

Furthermore, from our previous paper on SMS security [6], the following proofs of concepts are relevant for the threat scenarios “Attacks against secure channel device to mobile network”, “SIM-swapping Attacks” and “Authentication Data Attacks”:

- Passive radio channel interception of unencrypted SMS in 4G with disabled crypto.
- Downgrade attack of 4G network service by external attacker using the radio channel.
- Cryptographic attack on the 2G (GPRS) encryption algorithm GEA-1.
- Cryptographic attack on the 2G (GSM) encryption algorithm A5/1.
- Interception of SMS by operator or attacker with operator access rights in 4G.
- Interception of SMS by operator or attacker with operator access rights in 2G.
- SMS spoofing attack for attacker with operator access rights in 4G.
- SMS spoofing attack for attacker with operator access rights in 2G.
- SIM Swapping attack in the 4G network.



#	Threat Scenario	Proofs of Concept
1	Authentication data attacks.	• Interception of SMS by operator or attacker with operator access rights in 4G • Interception of SMS by operator or attacker with operator access rights in 2G
2	SIM-swapping attacks.	• SIM-swapping attack in the 4G network
3	Attacks on secure channels between the mobile network, the application backend and the application.	• Out of scope
4	Attacks against critical decisions functions.	• Out of scope
5	Attacks against secure channel device to mobile network.	• Passive radio channel interception of unencrypted SMS in 4G with disabled crypto • Downgrade attack of 4G network service by external attacker using radio channel • Cryptographic attack on the 2G (GPRS) encryption algorithm GEA-1 • Cryptographic attack on the 2G (GSM) encryption algorithm A5/1
6	Attacker with SMS access on device.	• SMS-OTP attack
7	Phishing attack to obtain SMS-OTP from user.	• SMS-OTP phishing attack
8	Attacker with network access on mobile device.	• Network-based attack
9	SIM-based phone number verification spoofing via malicious application.	• Out of scope

Table 10. Proofs of concept corresponding to the threat scenarios.

In the Appendix 3, the performed proofs-of-concept are presented, starting with a summary on the setup.

7 Regulations

The primary regulation governing phone number verification today is the Payment Services Directive 2 (PSD2) [22]. PSD2 is a European regulation that governs electronic payment services and their providers within the European Union (EU) and the European Economic Area (EEA) and that is relevant to authentication based on phone number verification. PSD2 came into effect on January 13, 2018, replacing the previous directive, PSD1.

PSD2 has several key objectives:

- Enhanced Security in Electronic Payments: It introduces Strong Customer Authentication (SCA), requiring at least two out of three authentication elements: something the user knows (e.g. password), something the user possesses (e.g. device), and something the user is (e.g. fingerprint).
- Third-Party Access to Bank Information: Banks are required to allow authorized third parties (Third Party Providers, TPPs) to access customer bank account information with customer consent. This facilitates the creation of new financial services, such as payment initiation and account information services.
- Consumer Protection: It enhances consumer protection in areas like transaction cost transparency and consumer rights in disputes over unauthorized payments.
- Transparency and Efficiency: It increases transparency in fees and the payment process, contributing to greater efficiency in the electronic payments market.

7.1 Is SMS-OTP PSD2 Compliant?

The answer is that SMS-OTP can be compatible with PSD2 requirements for Strong Customer Authentication (SCA), but there are specific conditions and considerations to ensure possible compliance.

- SMS-OTP can fulfill the possession element, since it demonstrates that the user has access to a registered mobile phone. For full SCA compliance, SMS-OTP must be combined with an element of knowledge (password or PIN) or inherence (biometric).
- Dynamic Linking was introduced in PSD2 to avoid attacks from social engineering. This aims to specifically link each transaction to its amount and the recipient of the payment. The SMS protocol is SS7, and since it is not a secure protocol, the European Banking Authority (EBA) indicates that SMS-OTP is not compatible with PSD2 [23] over the dynamic link, unless it is complemented with additional encryption or is transmitted over a secure channel.

7.2 Is SIM-based and Network-based Phone Number Verification PSD2 Compliant?

SIM-based and network-based phone number verification is not explicitly mentioned in the context of PSD2, but providers [24] assess that their network-based phone number verification solution is compatible with PSD2.

Based on the analysis in this paper and using a similar argument as for SMS-OTP, SIM-based phone number verification would constitute an attractive candidate for authentication to fulfill PSD2. SIM-based phone number verification demonstrates the possession of an element, the mobile phone, in a similar, and arguably even more secure, way compared to SMS-OTP. Furthermore, in contrast to SMS-OTP, social engineering attacks against codes are not applicable.

An advantage with network-based phone number verification is that social engineering attacks are not applicable in the same way as for SMS-OTP. The primary drawback of network-based phone number verification is its high susceptibility to local attacks from malicious applications.

8 Conclusions

This paper considers the most relevant methods for phone number verification that are critical in enhancing the security of financial applications and other sensitive services. A detailed analysis and comparison of the security properties and a threat analysis was applied to SMS-OTP, network-based and SIM-based phone number verification.

While SMS-OTP is widely adopted and accepted by the industry, it has significant vulnerabilities, including susceptibility to phishing attacks, local attacks, and security issues inherent to the SMS channel. Network-based phone number verification, although it mitigates phishing attacks, remains vulnerable to local attacks from malicious applications with standard privileges, as demonstrated in a proof of concept. In contrast, SIM-based authentication offers more robust security by leveraging the cryptographic measures inherent in SIM cards, significantly reducing vulnerability to local attacks. This makes SIM-based authentication an appealing future option when it becomes widely available.

With proofs of concept included, the practical feasibility of the potential attacks is demonstrated. It is important to point out that, even if there are possible attacks against an additional authentication factor, it is always recommended to use a second or multi-factor authentication method.

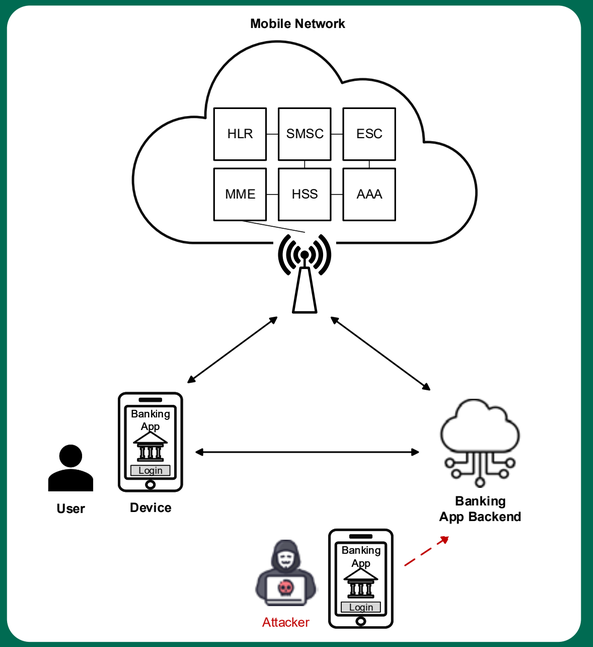


Figure 18. While the analyzed phone number verification methods have different challenges and strengths in terms of security and usability, using a second authentication factor is instrumental for a secure solution.

Appendices

Appendix 1

Security Properties Applicable to all Studied Methods

In this section the security properties that are identified to apply to SIM-based, SMS-OTP and network-based phone number verification are presented and illustrated. In this case, the objective is to present the methods and main properties in an as simple way as possible. To achieve this, some simplifications and abstractions have been made. As an example, the mobile network consists of a number of entities that are involved with different roles and functions. However, since the focus of this paper is not core network security, the mobile network is illustrated as a single entity for simplicity.

Security Property Identifier	Definition of Required Security Property	Security Solution
[SECURE_CORE_NETWORK]	Core network security ensures that the various entities involved in phone number verification methods are interconnected through secure channels, with each entity maintaining a sufficient level of security.	Mobile and core network security solutions.
[SECURE_CHANNEL_MOBILE_DEVICE_TO_MOBILE_NETWORK]	The channel between the mobile device and mobile network is secure.	Based on the same principles and standards that define the secure authentication between the device and the mobile network.
[SECURE_DEVICE_NETWORK_AUTHENTICATION]	The mobile network device authentication procedure is secure and implemented securely.	Including, but not restricted to, the EAP-AKA security solution, part of the mobile network implementation.
[SECURE_CHANNEL_MOBILE_NETWORK_TO_APP_BACKEND]	The channel between the mobile network and the application backend is secure.	A solution for this is using public key infraestructure together with TLS to achieve a secure channel.
[SECURE_CHANNEL_APP_TO_APP_BACKEND]	The channel between the application on the device and the application backend is secure. Note that the phone number verification can be seen to authenticate this channel with an additional method.	A solution for this is public key infrastructure together with e.g. TLS to achieve a secure channel.

Table 11 (1/2). Security properties applicable to all studied methods for phone number verification.

Security Property Identifier	Definition of Required Security Property	Security Solution
[SECURE_DATA_ MANAGEMENT_ OPERATOR]	The mobile operator manages data used for phone number verification in a secure way.	For example, access to the data critical for network-based phone number verification is restricted using strong isolation mechanisms and data is erased after use. The data should be managed as a sensitive security asset. Similar solutions apply for SMS-OTP, for example erasing the SMS-OTP from the SMS-Gateway when sent, and that access to the SMS-OTP in all points of the network is protected during the validity period of the SMS-OTP. Similarly, for SIM-based phone number verification, the operator‘s data management is crucial for the relation between the symmetric keys and the phone number.
[SECURE_APP_ BACKEND]	The application backend security manages access control, authentication, authorization, separation between users and data, input validation and processing, and user data etc. In particular, the data used for phone number verification is managed securely.	State-of-the-art backend security solutions. In particular, for phone number verification, similar solutions as for the operator for the secure handling of data network-based, SMS-OTP and SIM-based phone number verification can be used, see [SECURE_DATA_ MANAGEMENT_OPERATOR].
[SECURE_VALUE_ COMPARISON]	The mobile operator, or application backend, performs a secure comparison of the reference values with the obtained values from the phone number verification.	For network-based phone number verification, secure comparison of the connection data, or phone number, for the mobile network data connection with the connection data or phone number to the application backend. Secure comparison of potential tokens used. For SMS-OTP, verification of the SMS-OTP is securely performed, for example considering validity time of the OTP, possible brute force and timing attacks. For SIM-based phone number verification, the complete validation of the access token, the application ID and the OperatorToken is correctly performed by the corresponding entity.

Table 11 (2/2). Security properties applicable to all studied methods for phone number verification.

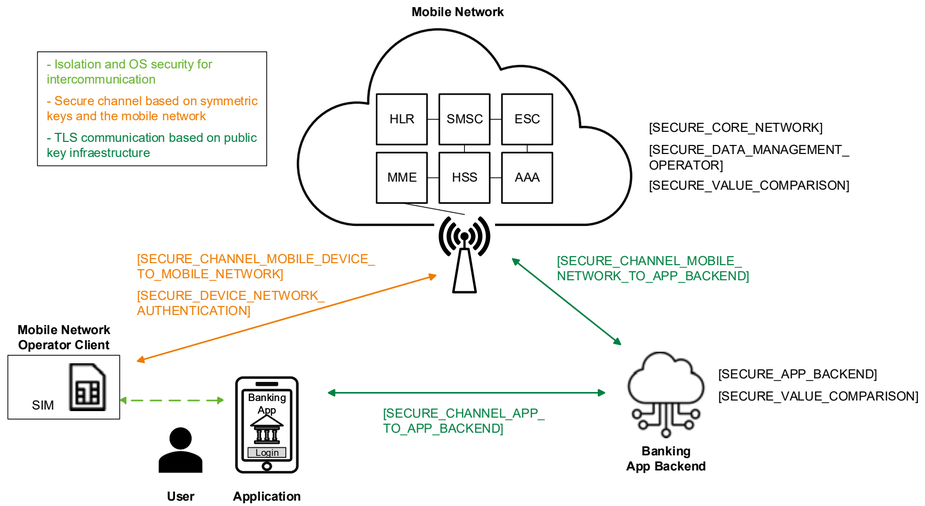


Figure 19. Security properties applicable to all studied methods.

Appendix 2

In this appendix, threat scenarios that are common for all the phone number verification methods are given.

Threat Scenario 5: Attacks on Authentication Data

Secure data management of SIM information, keys, and user details is critical for the authentication process, as well as the proper handling of this information. This holds for both the data management by the entities of the mobile network and for the application backend.

An attacker with privileges to modify information critical for authentication in the mobile network, for example through a network attack, can affect the outcome of the authentication by modifying data.

For example, an attacker with operator-level privileges could modify or retrieve information within the core network to influence the outcome of the authentication in their favor, for example by:

- Modifying the information that network-based phone number verification is using, such as the IP-number, or the phone number for verification.
- Retrieving the SMS-OTP value.
- Affecting the authentication procedure or leaking the tokens for SIM-based phone number verification.

An attacker with the necessary privileges in the application backend can potentially exploit the victim's resources directly. Additionally, they can manipulate the authentication implementation, configuration, or authentication data to gain access, similar to an attacker with operator-level privileges.

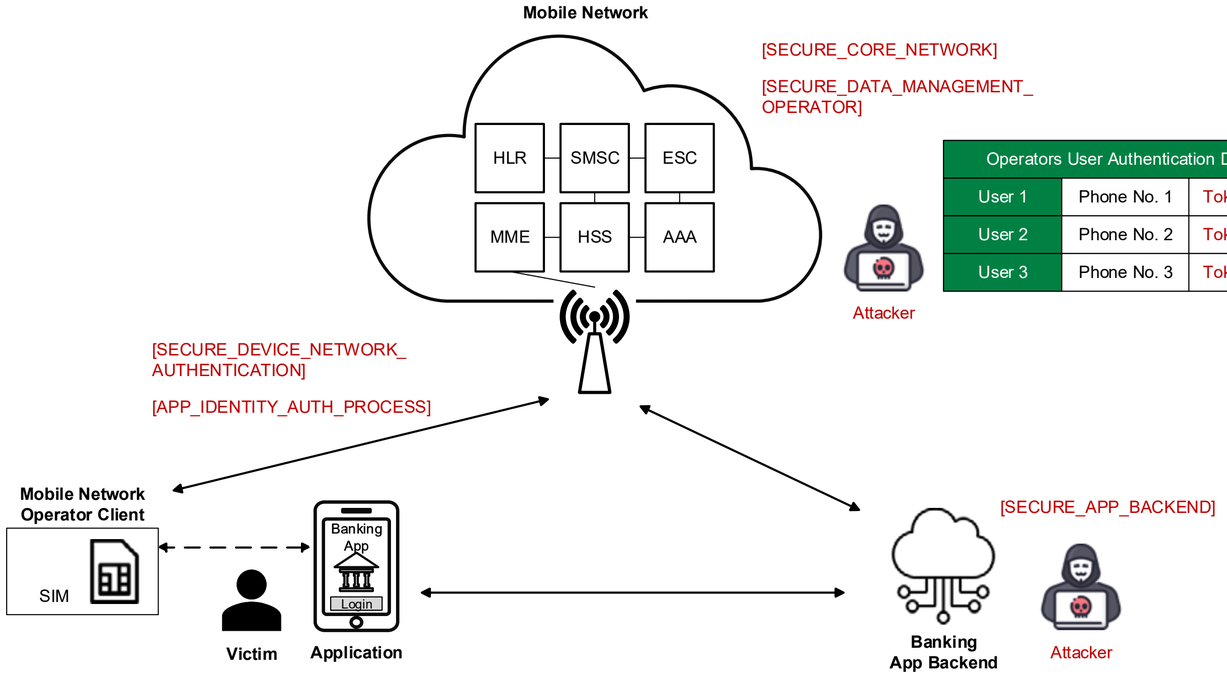


Figure 20. Illustration of the threat scenario of attacks against authentication data.

Threat Scenario 6: SIM-Swapping Attacks

In a SIM-swapping attack, the security foundation for the device authentication is lost by an attack of the coupling between the phone number and the device. After the attack, another device can be authenticated as having the victim's phone number.

In the SIM-swapping attack, the attacker is able to modify the association of the phone number with another SIM-card. This could be done by a social engineering attack, by non-trustworthy personnel at the operator or by attacks on the operators' systems.

SIM-swapping attacks can be seen as an important special case of the threats described in Threat scenario 1.

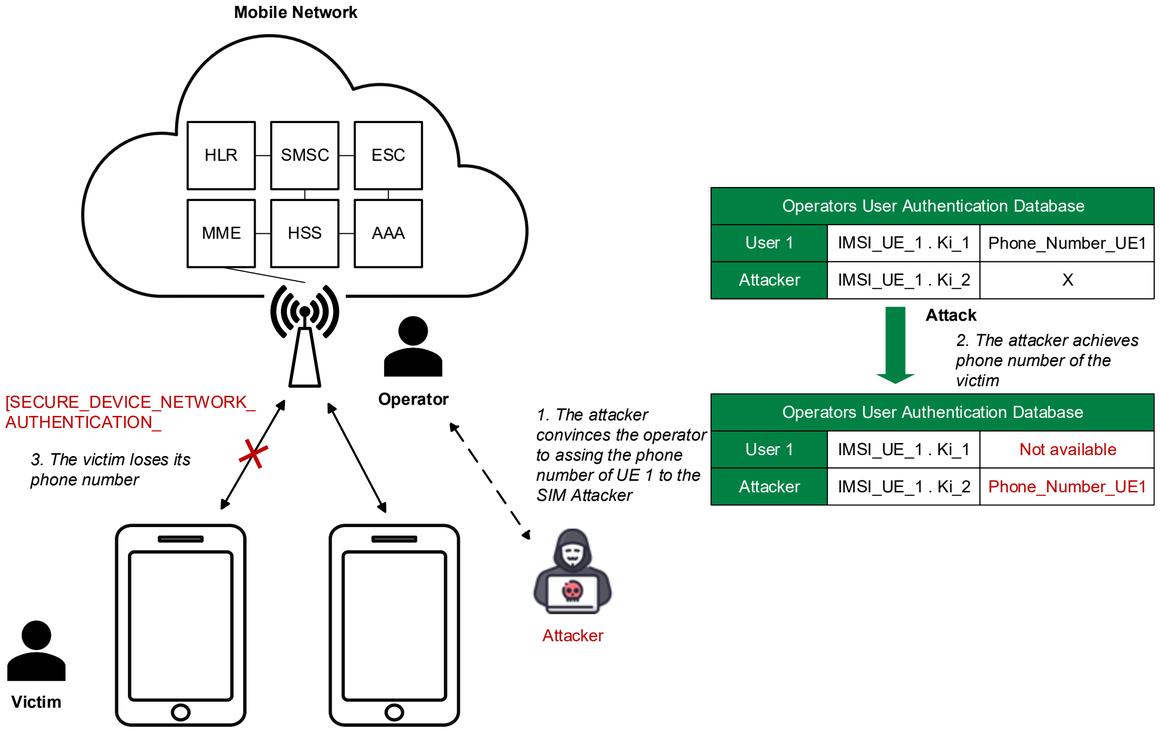


Figure 21. SIM-swapping attack where the attacker convinces the operator to assign the victim's phone number to the attacker´s SIM-card.

Threat Scenario 7: Attacks on the Secure Channels Between the Mobile Network, the Application Backend and the Application

The secure channels between the application and the application backend, and the application backend and the mobile network are crucial for all the phone number verification methods.

Based on the channel and method the threat scenarios are divided into:

- Modifying the result message of network-based authentication to facilitate authentication of the attacker to the victim´s resources.
- Leaking the SMS-OTP value to a device controlled by the attacker, permitting the attacker to enter the SMS-OTP from its own device and this way getting authenticated to the victim´s resources.
- Leaking or setting the tokens for SIM-based authentication, permitting the attacker to authenticate with the tokens from its own device and this way getting authenticated to the victim´s resources.

Attacks on the Secure Channel Between the Application and the Application Backend

An attacker that compromises the security of the channel between application on the device and the application backend could act as a man-in-the-middle to gain access to the victim´s resources.

In one attack path, the attacker could simply intercept the information on the channel, this way getting access to those resources.

In other attack scenarios, the attacker could perform an active role. This includes obtaining authentication information, for example:

- The authentication token for SIM-based authentication, which the attacker could use to get access to the victim´s resources from another device.
- The SMS-OTP value, which the attacker also could use to gain access to the victim´s resources from another device.

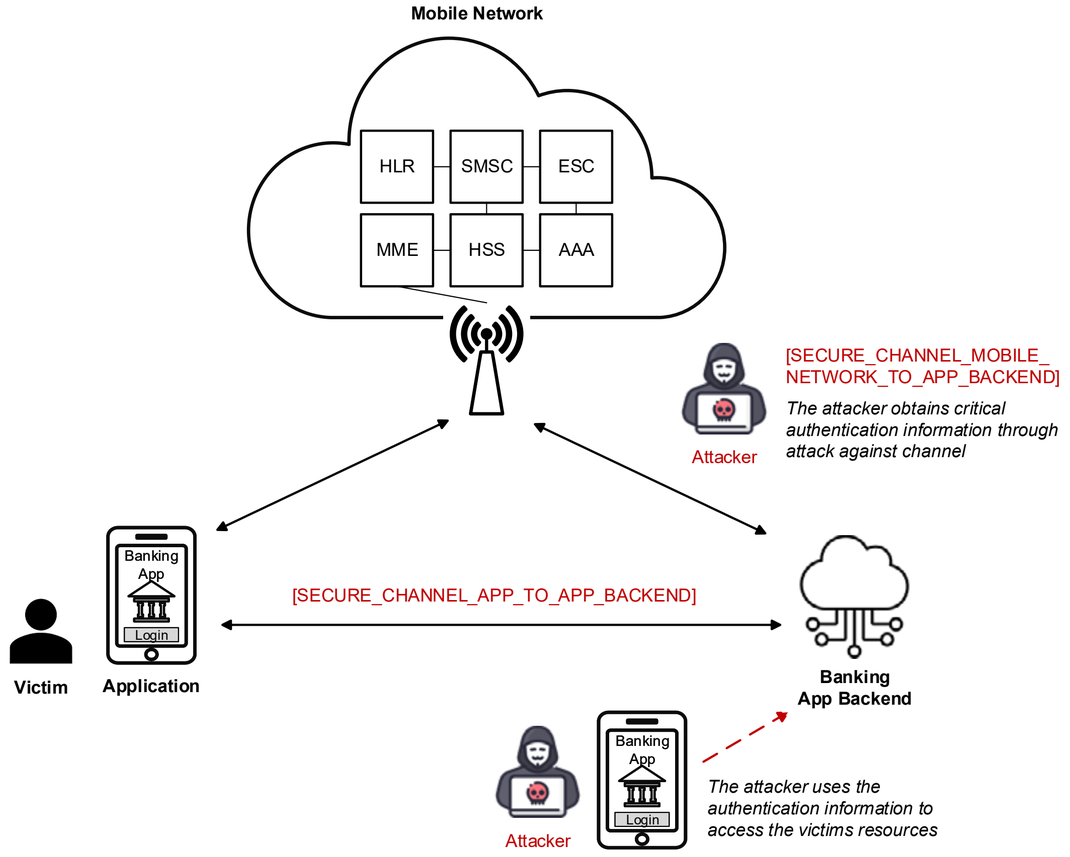


Figure 22. Illustration of threat scenarios of attacks against the secure channels between the mobile network, the application backend and the application.

Threat Scenario 8: Attacks Against Critical Decision Functions

In implementations for phone number verification, there are critical decision points, where received information is compared to reference information. One clear example of this is the comparison of the OTP value that was generated and sent through the SMS-channel, with the OTP that is later received through the user interaction with the application.

These comparison functions, including the correct actions based on the comparison, are critical for the security. If not properly implemented, an attacker could be authenticated, even without correct information. In fact, there are a number of attacks against this type of decision functions that should be considered.

One example of such attacks is replay attacks, where, for example, older OTPs are reused for access. Another example is timing attacks, where the attacker can take advantage of correlations between execution time of a comparison and the data values in the comparison. Using attack strategies involving repeated guessing and observations, the attacker can, if the comparison function is vulnerable to this type of attack, infer the correct value and succeed in the attack.

Another relevant attack on the critical decision functions is brute-force attacks, where the attacker repeatedly tests different values until finding the correct one. This can be a feasible attack against for example SMS-OTP, especially if they are not generated with high enough entropy and if there are not sufficiently efficient methods to detect and mitigate such attacks. Versions of these attacks are also possible against SIM-based phone number verification regarding the access token and OperatorToken.

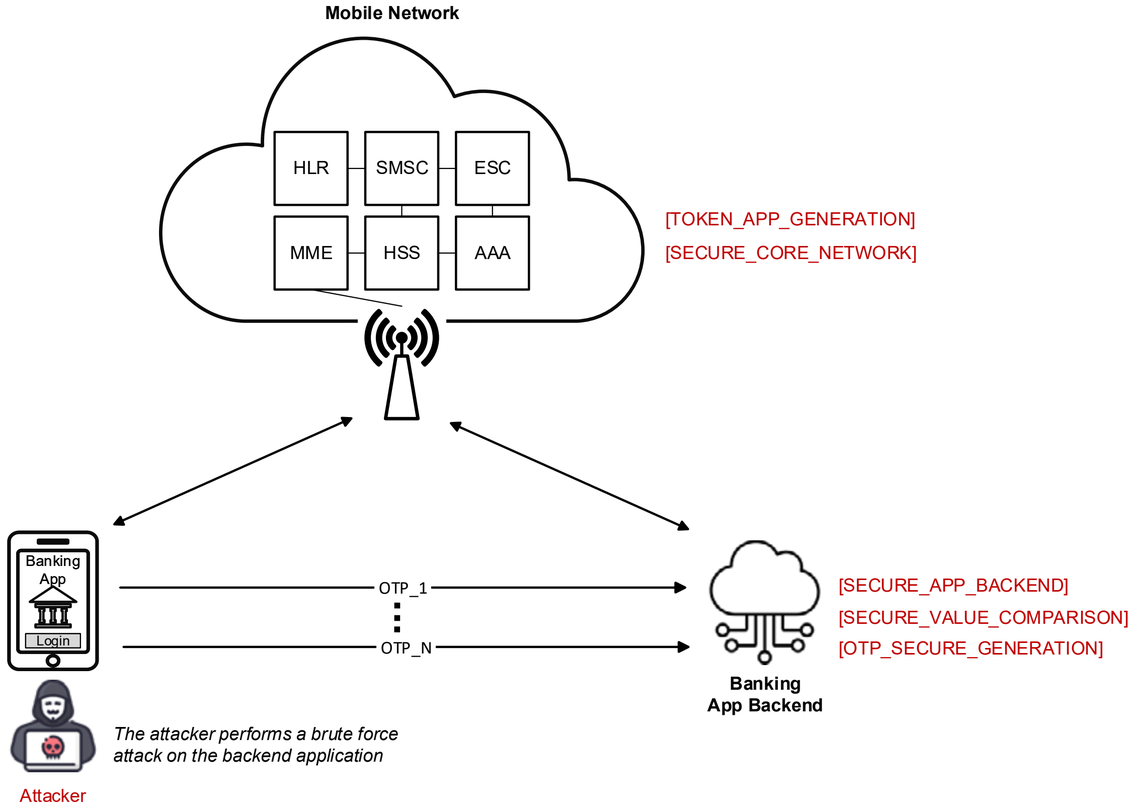


Figure 23. Brute-force attack against SMS-OTP.

Threat Scenario 9: Attacks Against the Secure Channel Between the Device to the Mobile Network

A key element in the security of phone number verification based on SMS-OTP and SIM is the confidentiality of the OTP value and the tokens used for the authentication. While confidentiality is important, especially for the case of SMS-OTP, it is also worth noting that if a short validity time is used, an attacker would need to retrieve the OTP value and perform the attack in real-time, which raises the complexity of the attack. Even so, this is a relevant threat scenario.

In Threat Scenario 1, the interception of authentication data within the core network was addressed. Additionally, the secure channel between the application backend and the mobile network was covered in Threat Scenarios 3, 4, and 5. However, it is also worth considering attacks on the radio channel from the mobile network to the mobile device.

In most cases, the radio channel is securely encrypted, but there are still a number of threats to the security on the channel that remain. This includes the case where 2G is still supported. Attacks against all generations of mobile networks are then vulnerable, if they have implemented a fallback solution to 2G. For example, an attacker can relatively easily perform a downgrade attack to change 5G service to 2G service, which makes the user vulnerable to the threat scenarios for 2G. The security functions of 2G have vulnerabilities and practical attacks against the security are possible, for example to retrieve SMS-OTP.

The threat scenarios involve active man-in-the-middle attacks, but also passive interception attacks. The passive interception attacks can be based on either cryptographic attack against the legacy cryptographic algorithms in 2G, or on the threat scenario of the radio channel being configured without security. The operator has the possibility to control which cryptographic algorithms that are used, and also the possibility to turn off the protection of SMS. In such a scenario an attacker in radio range could intercept the messages in clear text.

While a downgrade attack to 2G results in an insecure channel, the authentication mechanism on which SIM-based authentication is based, is unavailable for 2G. This means that rather than an attack retrieving, for example, authentication tokens for SIM-based phone number verification, a downgrade attack to 2G implies a downgrade to other authentication methods, such as SMS-OTP.

In this threat scenario, attacks on the device authentication mechanism to the mobile network is also included. This authentication mechanism, typically EAP-AKA, is fundamental for device authentication and to create a secure connection to the mobile network.

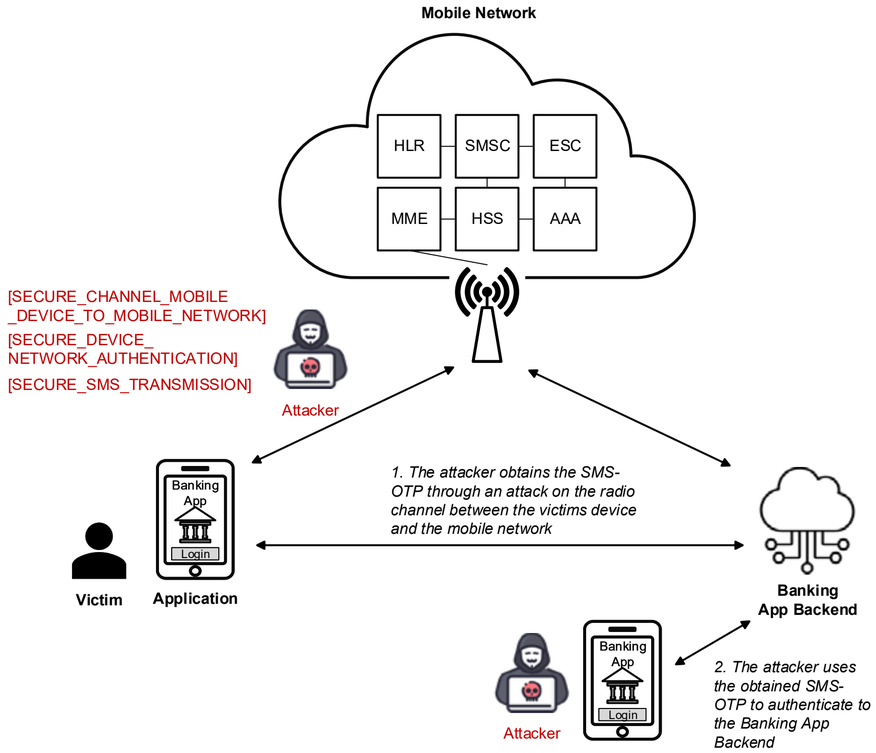


Figure 24. Attack against the secure channel between the device to the mobile network.

Appendix 3

In this appendix, the proofs of concept that are not described in [6] are presented.

PoC Setup

In the realization of the PoCs, both legal and practical aspects need to be considered. All PoCs have been performed in a controlled environment in a DEKRA laboratory. This means that DEKRA has set up its own mobile infrastructure to perform the attacks. This has been in an isolated environment, not connected to, or interfering with, public mobile networks.

Network Simulated	4G
Develop Apps Program	Android Studio 2023 3.1
Network	Amarisoft Callbox Classic
User Equipment	Standard mobile phones have been used: Samsung Galaxy S22, Xiaomi Redmi 9

Table 12. Equipment used in the execution of the proofs-of-concept

The objective was to create a testing environment that reflects real-world scenarios.

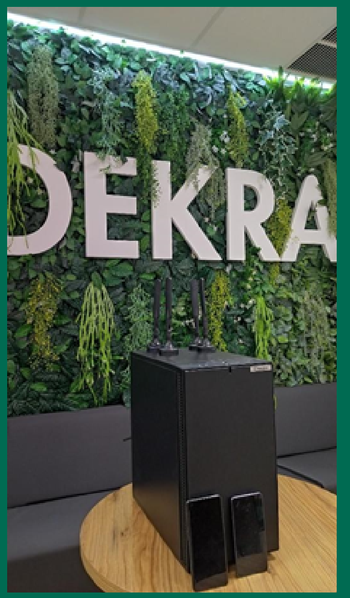


Figure 25. Amarisoft Classic with the two phones.

Network-based Attack

Objective

Malicious application, which implements network requests, letting the attacker relay network traffic through the victim´s device. The fact that the attack is performed from the victim‘s phone, and the phone number verification will verify that it is the victim‘s phone, makes the attack possible.

Setup

In this PoC, both mobile devices were connected using the simulated operator and mobile network.

In this case, the devices were connected to our simulated network for simplicity, but in order to reproduce the PoC there is no need to use a malicious base station nor connecting the devices to the same base station.

The setup to this PoC is:

- PC: To configure the 4G network via SSH.
- 4G network (Amarisoft Callbox Classic): Simulates a real operator network and a base station.
- Two mobile phones: Simulating real communication.
- Two writeable SIM cards: To connect the devices to the base station.
- Banking app: Simulates a service that contains a mobile application and a backend app.

Figure Conceptual: Figure 17

Figure Simulated:

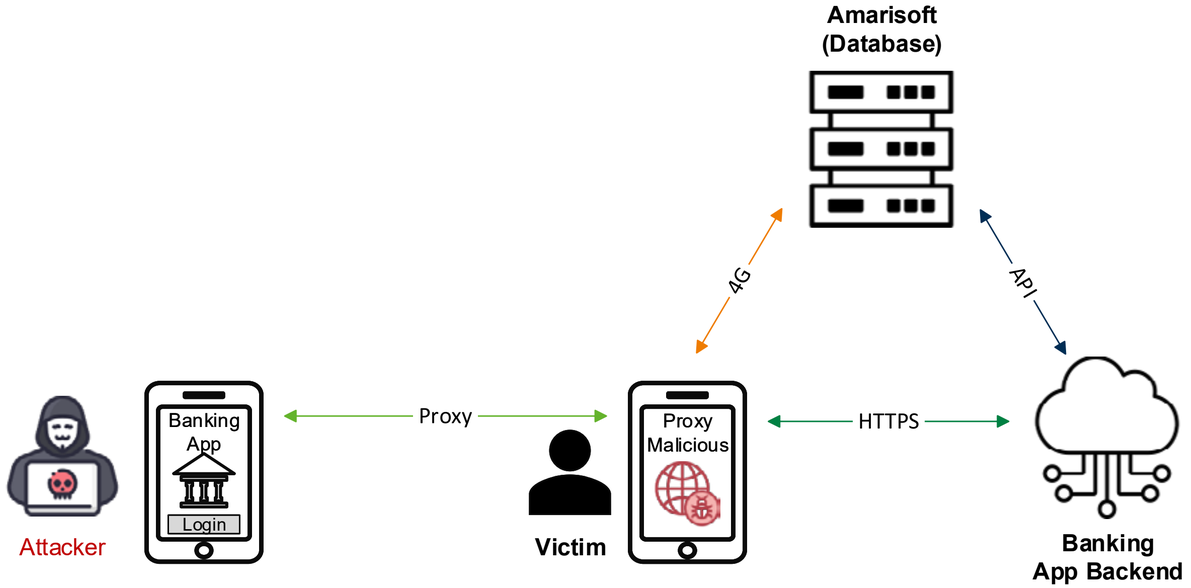


Figure 26. Simulated scenario of a network-based attack.

Configuration of Setup

The configuration to this PoC is:

- **Phone 1 (Attacker)**
  - Configuration SIM 1:
    - IMSI: 0010101023456789
    - Number Phone: 600
  - IP Address: 192.168.4.6
  - Installed Banking App

- **Phone 2 (Victim)**
  - Configuration SIM 2:
    - IMSI: 0010101023456788
    - Number Phone: 601
  - IP Address: 192.168.4.2
- IP Adress Phone 2 (Attacker): 192.168.4.6
- Internet and IMS APN configured in both phones

Procedure

The procedure of this PoC is:

1. Launch the bank application
2. Log in with your real number
3. Check the answer
4. Log in with the attacked number
5. Check the answer
6. Install the proxy malicious in the attacked phone.
7. Configure the Access Point Name (APN) on the attacker's phone
8. Log in with the attacked number
9. Check the answer

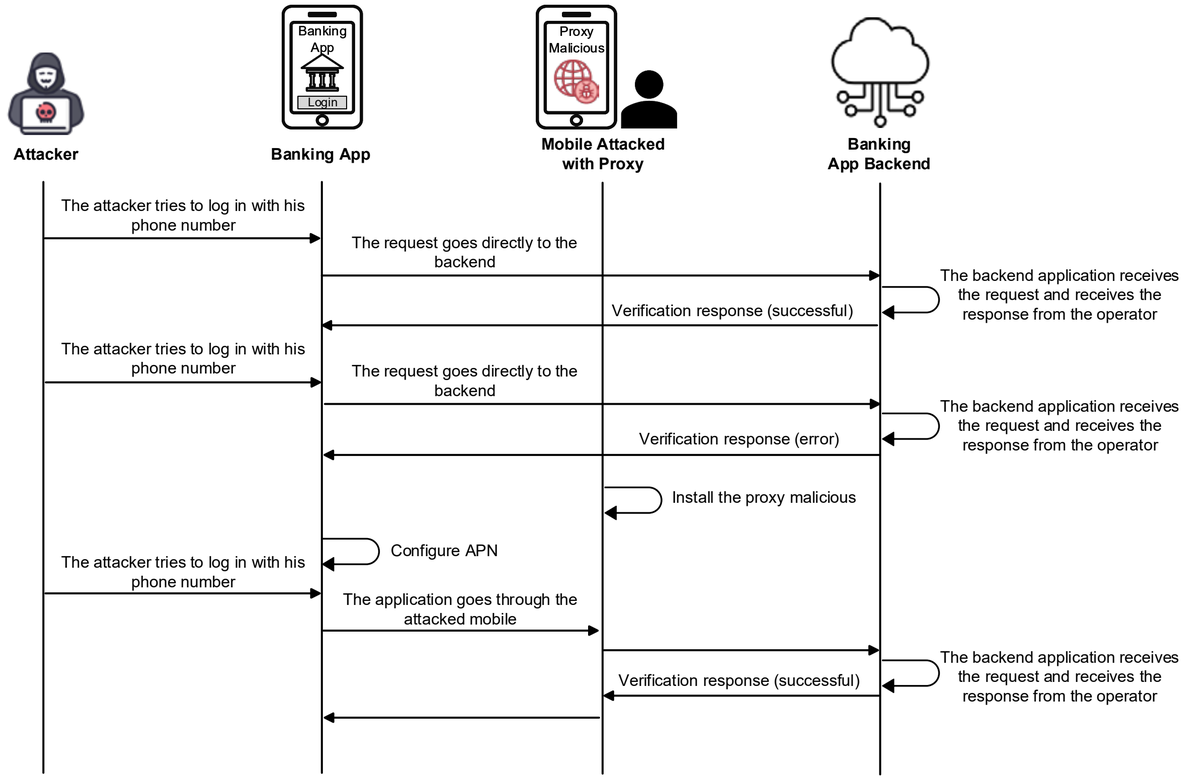


Figure 27. Procedure of the network-based attack proof of concept.

Results

The attacker launches the bank application.

Figure 28. Login to banking application.

The attacker verifies that he can log in with his phone number.

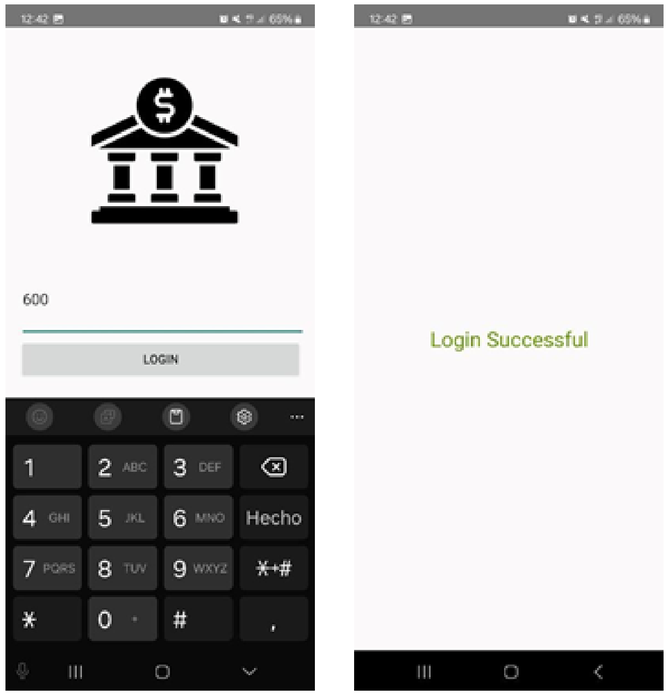


Figure 29. Successful login with the attacker's phone number.

The attacker tries to log in with someone else's phone number, but the login fails.

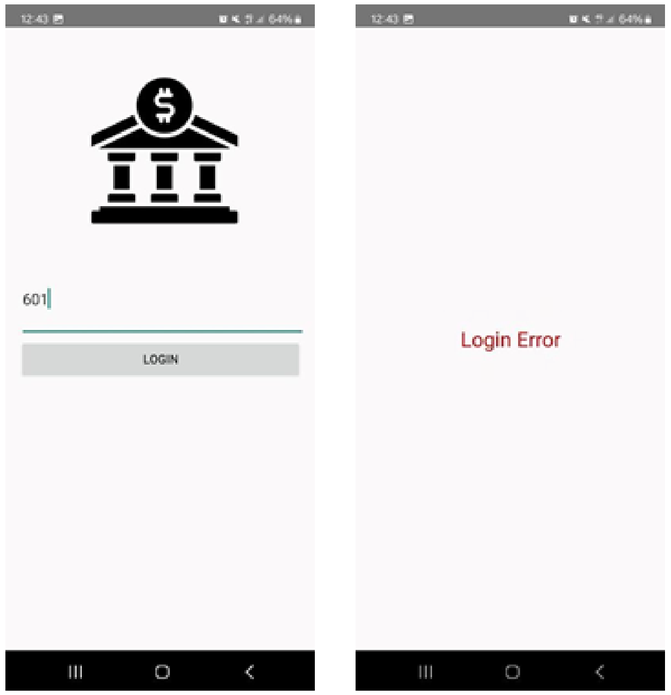


Figure 30. Non-successful login with the victim's phone number.

The backend of the application received two login attempts from two numbers. In the first attempt, the access is correct (response code 200), and in the second attempt it is incorrect (response code 300).

The first attempt is successful because the operator verified that the given number matches the IP of the request.

In the next step, the malicious proxy application is installed on the victim's phone. This application can be installed, for example, via a malicious link.

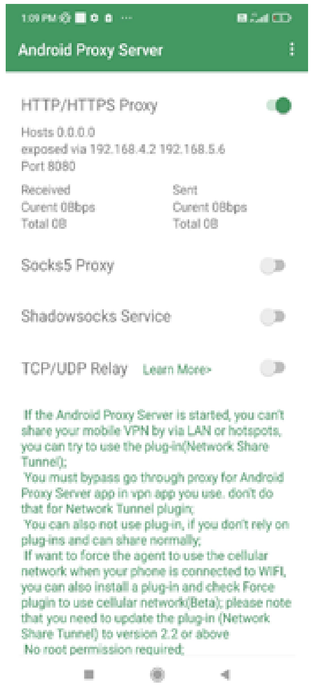


Figure 31. Configuration proxy server on the victim's phone.

The attacker configures the APN, indicating the IP-address of the attacked phone as a proxy.

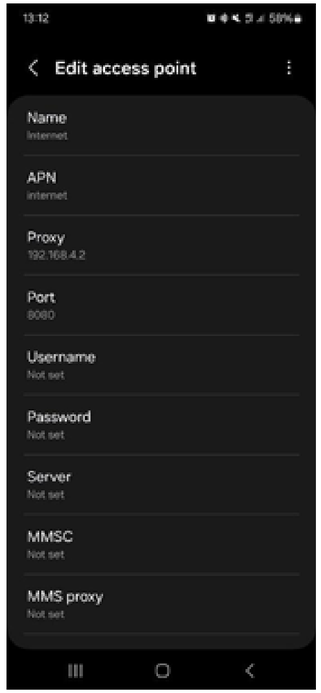


Figure 32. Configuration APN on the attacker's mobile.

The attacker tries to log in with the phone number of the victim and in this case the request passes through the proxy (on the attacked mobile). Therefore, the server infers that the request is made by the attacked phone.

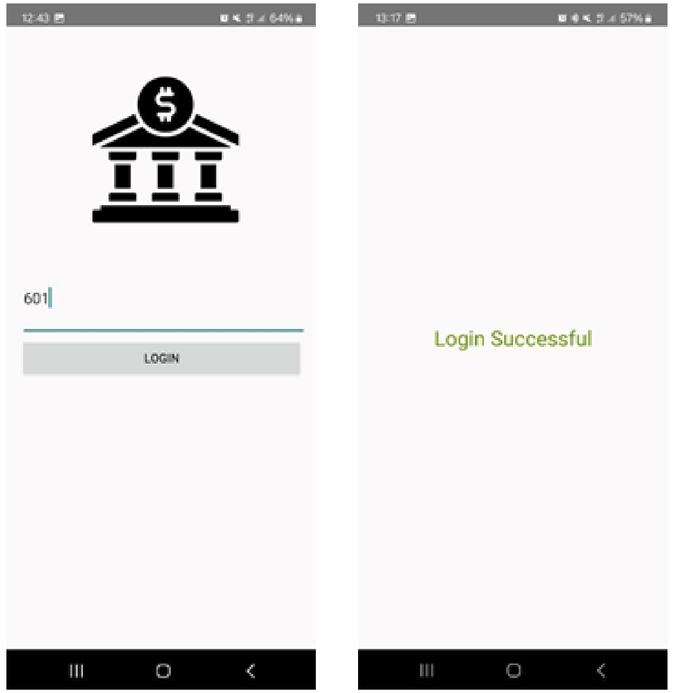


Figure 33. Successful login with the victim's phone number.

Conclusions

This PoC has illustrated a vulnerability with network-based phone number verification. A malicious actor can use an application as a proxy on the targeted phone, in order to carry out the attack. The victim is usually not aware of the attack.

SMS-OTP Attack

Objective

The objective of this PoC is to demonstrate the risk of a user accepting SMS read privileges to an application, which can lead to a malicious application relaying SMS-OTPs to an attacker that can access for example a banking application.

Setup

The setup to this PoC is:

- PC: To configure the 4G network via SSH.
- 4G network (Amarisoft Callbox Classic): Simulates a real mobile network and a base station.
- Two mobile phones: Simulating real communication.
- Two writeable SIM cards: To be able to connect them to the base station.
- Banking app: Simulates a service that contains a mobile application and a backend app.

Figure Conceptual: Figure 15

Figure Simulated:

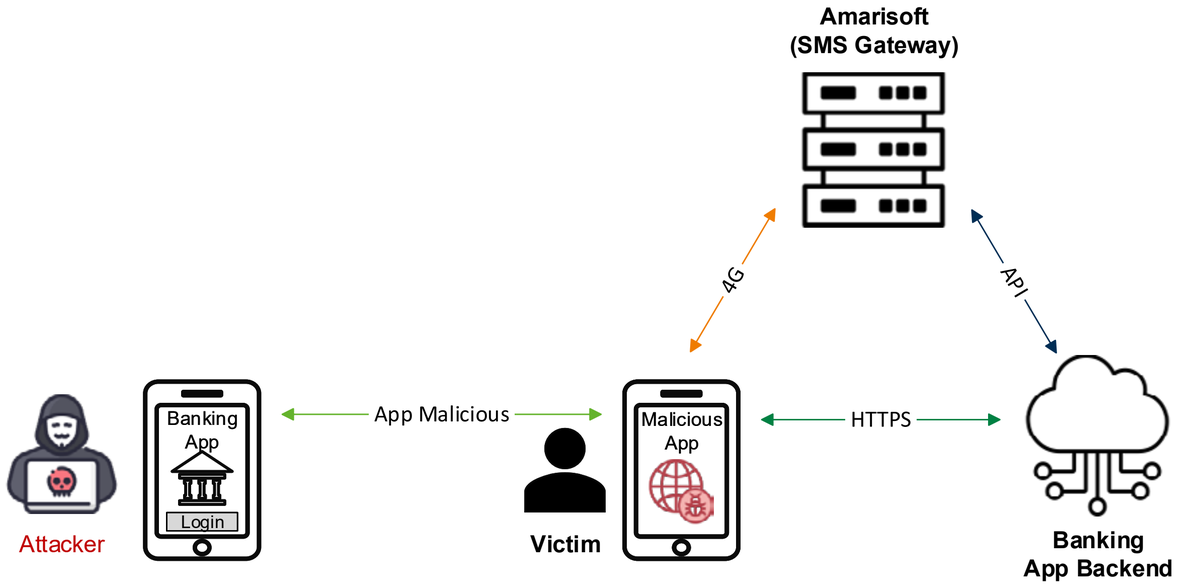


Figure 34. Simulated scenario of the SMS-OTP Attack proof of concept.

Configuration of Setup

The configuration to this PoC is:

- **Phone 1 (Attacker)**
  - Configuration SIM 1:
    - IMSI: 0010101023456789
    - Number Phone: 600
  - IP Address: 192.168.4.6
  - Installed Banking App

- **Phone 2 (Victim)**
  - Configuration SIM 2:
    - IMSI: 0010101023456788
    - Number Phone: 601
  - IP Address: 192.168.4.2
- IP Adress Phone 2 (Attacker): 192.168.4.6
- Internet and IMS APN configured in both phones

Procedure

The procedure of this PoC is:

1. Install the malicious SMS-reading application in the attacked phone
2. Install the SMS receiving application of the other application on the attacking mobile phone
3. Log in with the attacked number*
4. Check SMS OTP in the SMS receiving application*
5. Enter the SMS in the bank application*

** These steps are not shown in the PoC and correspond to a typical bank application login.*

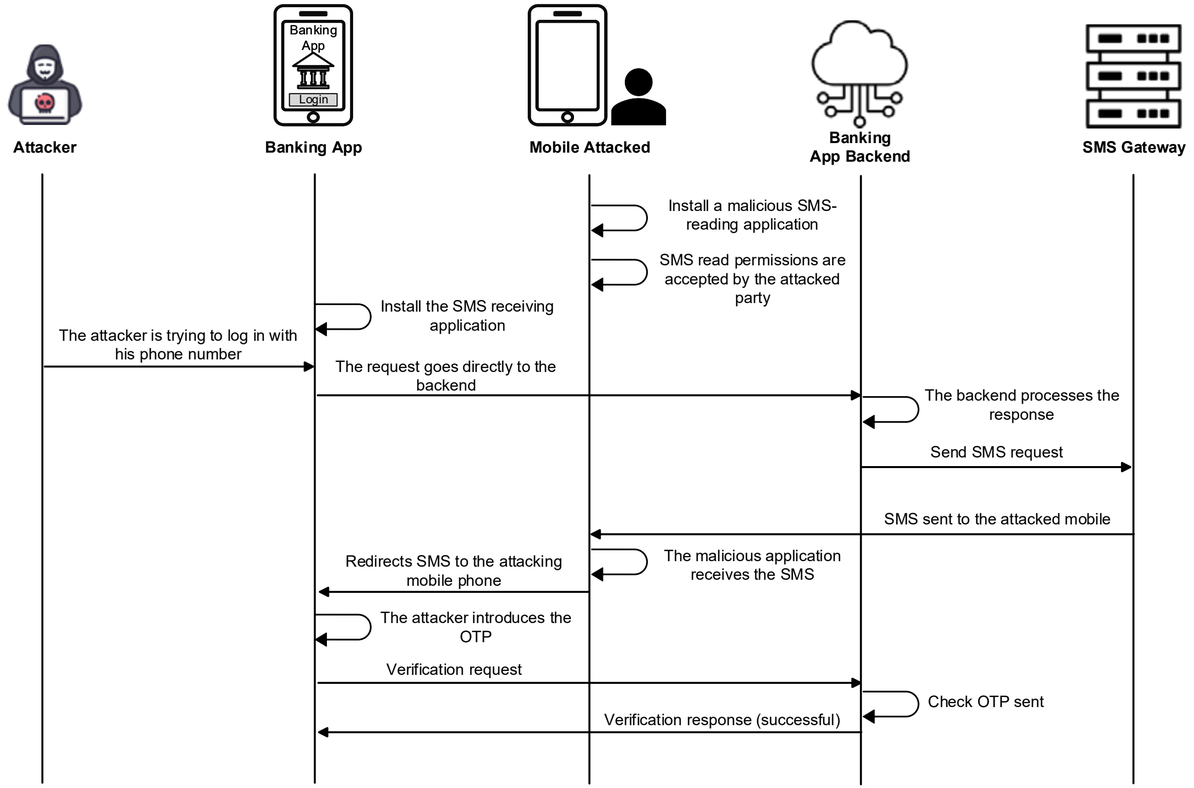


Figure 35. Procedure of the SMS-OTP attack proof of concept.

Results

The malicious SMS-reading application is installed on the attacked phone. This application can be installed, for example, via a malicious link.

Once the application is installed, the user can access and accept the permissions the application needs to function.

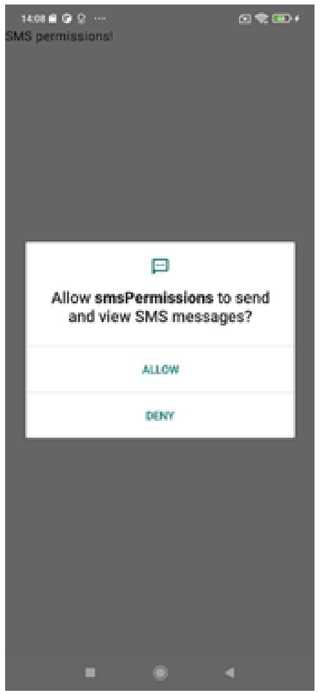


Figure 36. SMS permissions acceptance window of the malicious application.

If the victim accepts SMS read permissions, the attacker will have access to all messages and will be able to perform the attack.

Conclusions

This PoC demonstrates how a small user failure can breach the security against SMS-OTP attacks. Most users, when installing an application, accept all permissions without carefully reading and understanding the security implications.

SMS-OTP Phishing

Objective

The objective of this PoC is to demonstrate the risk that a user of a service using SMS-OTP could fall victim to a phishing attack with the aim of stealing their OTP and the attacker gaining access to their account.

Setup

The setup to this PoC is:

- PC: To configure the 4G network via SSH.
- 4G network (Amarisoft Callbox Classic): Simulates a real mobile network and a base station.
- Two mobile phones: Simulating real communication.
- Two writeable SIM cards: To be able to connect them to the base station.
- Banking app: Simulates a service that contains a mobile application and a backend app.

Figure Conceptual: Figure 16

Figure Simulated:

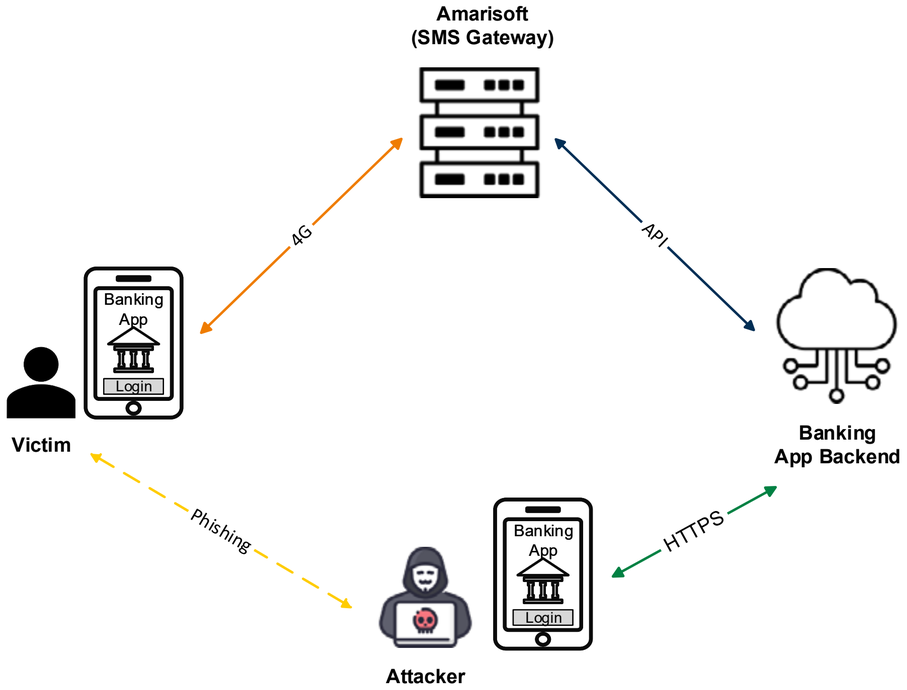


Figure 37. Simulated scenario of the SMS-OTP phishing attack.

Configuration of Setup

The configuration to this PoC is:

- **Phone 1 (Attacker)**
  - Configuration SIM 1:
    - IMSI: 0010101023456789
    - Number Phone: 600
  - IP Address: 192.168.4.6
  - Installed Banking App

- **Phone 2 (Victim)**
  - Configuration SIM 2:
    - IMSI: 0010101023456788
    - Number Phone: 601
  - IP Address: 192.168.4.2
- IP Adress Phone 2 (Attacker): 192.168.4.6
- Internet and IMS APN configured in both phones

Procedure

The procedure of this PoC is:

1. Login with the attacked number in the banking application.
2. The victim will receive the SMS-OTP.
3. Perform a phishing attack to obtain the OTP.
4. The victim is deceived to provide the SMS-OTP to the attacker.
5. Enter the SMS-OTP in the bank application.

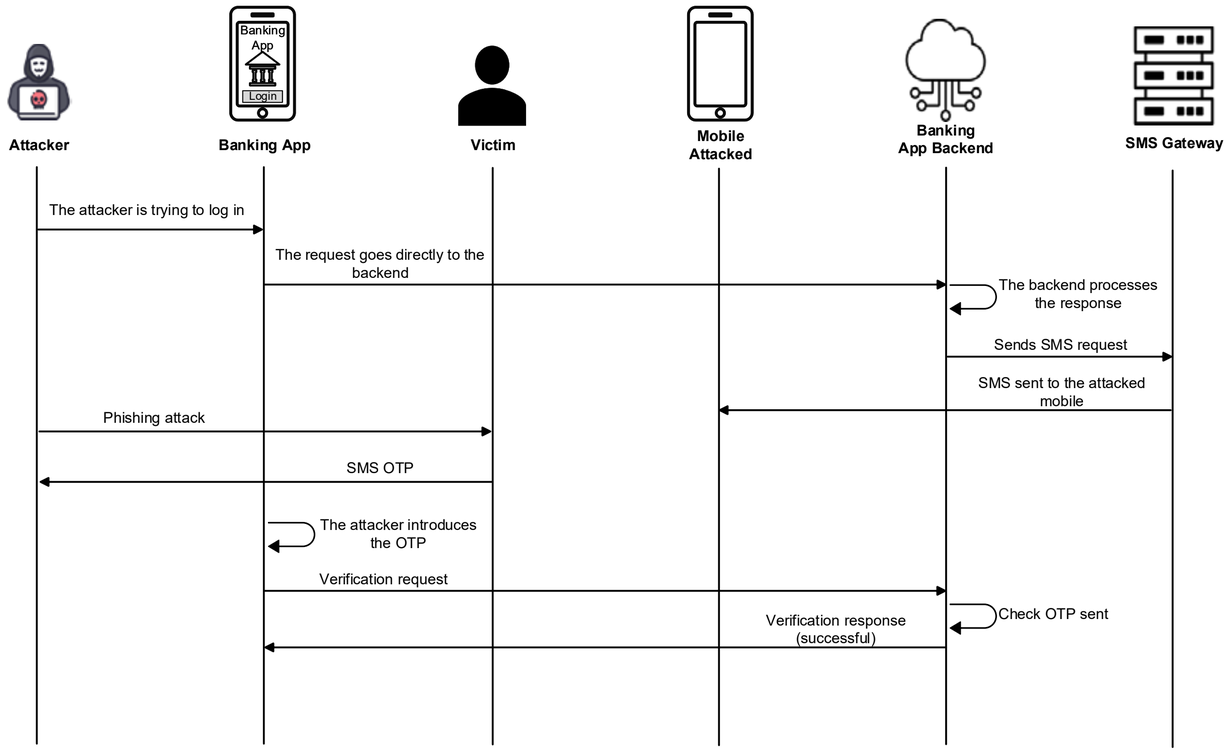


Figure 38. Procedure of the SMS-OTP phishing attack.

Results

The attacker launches the banking application.

Figure 39. Login to banking application.

The attacker initiates login with the phone number of the victim.

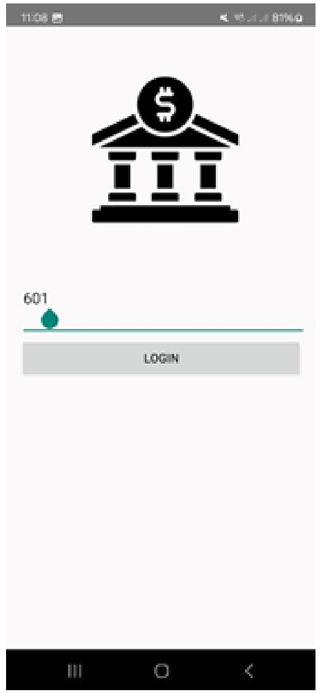


Figure 40. Login with the victim's number.

The bank requests the attacker to enter the SMS-OTP.

Figure 41. Entering the OTP sent by the backend.

The SMS-OTP will be received on the victim's mobile phone. The attacker performs a phishing attack. There are several ways that an attacker can perform the phishing attack. For example, the attacker can call the victim and impersonate the bank to convince the victim to provide the OTP. Another option is to use an email or SMS to impersonate the bank and ask for the OTP.

Results

Once the attacker has obtained the OTP, the attacker accesses the banking application.

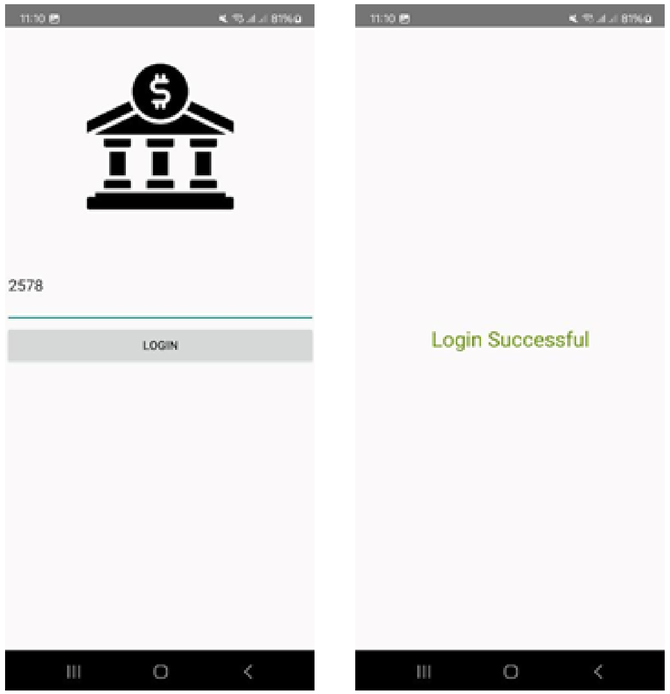


Figure 42. Successful login with the victim's phone number.

Conclusions

This PoC demonstrates that SMS-OTP security is vulnerable to phishing attacks. Many users have been exposed to a phishing attack at least once in their lifetime. Phishing attacks are easy to carry out and the consequences can be very serious.

Appendix 4

SMS-OTP in relation to OTP based on HOTP and TOTP

The basic underlying assumption for the authentication based on HOTP and TOTP is that both parties share the same secret used in the process of generating the OTP and keep track of the variable input. SMS-OTP works differently in that there is no shared key and independent computation of the OTP for both sides. This means that only some aspects of the security analysis of HOTP and TOTP are useful for the security analysis of the SMS-OTP method, and in particular the security of SMS-OTP cannot be based on security of HOTP and TOTP. For the SMS-OTP case, instead of generating the OTP, a separate secure channel is needed to deliver the OTP to the user, so that the user can use it for authentication.

The difference of the SMS-OTP method compared to HOTP and TOTP implies that

- The separate channel used for provisioning the user with the OTP is SMS through the mobile network. This requires knowledge of the phone number, which is why SMS-OTP can be interpreted as a phone number verification and the access to the device.
- The whole OTP generation procedure, that the OTP is actually generated securely, with sufficient entropy, unique keys for each user and other factors to consider, is out of the user control for the typical SMS-OTP use case.

In fact, there are applications for mobile devices and tokens that do implement HOTP and TOTP, but the security assumptions are different.

Abbreviations

APN: Access Point Name

HOTP: HMAC-based One Time Password

IMSI: International Mobile Subscriber Identity

OTP: One Time Password

PKI: Public Key Infrastructure

PoC: Proof of Concept

PSD: Payment Services Directive

SCA: Strong Customer Authentication

SIM: Subscriber Identity Module

SMS: Short Message System

TOTP: Time-based One Time Password

References

[1] E. D. Hardt, “RFC 6749 - The OAuth 2.0 Authorization Framework,” 2012.

[2] E. Tatlı, “Cracking More Password Hashes With Patterns,” IEEE, 2015.

[3] T. A. West, “Weak Password Policies:,” CISSE, 2020.

[4] S. M. N. V.-R. J. T. José Miguel Moreno, “Your Code is 0000: An Analysis of the Disposable Phone Numbers Ecosystem,” IEEE, 2023.

[5] “Official Document TS.43 - Service Entitlement Configuration Version 11.0,” GSMA, 01 April 2024.

[6] F. G. J. N. J. J. Anders Möller, “SMS doesn‘t stand for secure messaaging services: SMS (In)Security revisited,” DEKRA, 2023.

[7] M. E. S. C. Nicole Allen, “Mobile Devices in 2021,” 25 November 2021. [Online]. Available: https://www.cyberdefensemagazine.com/the-5-most/.

[8] Y. N. Y. F. A. B. Zeyu Lei, “On the Insecurity of SMS One-Time Password Messages against Local Attackers in Modern Mobile Devices,” NDSS, 2021.

[9] L. Bicchierai, “Vice,” 3 August 2018. [Online]. Available: https://www.vice.com/en/article/3ky5a5/criminals-recruit-telecom-employees-sim-swapping-port-out-scam.

[10] S. Gatlan, “Bleeping Computer,” 26 February 2021. [Online]. Available: https://www.bleepingcomputer.com/news/security/t-mobile-discloses-data-breach-after-sim-swapping-attacks/.

[11] X. Landen, “Newsweek,” 12 May 2022. [Online]. Available: https://www.newsweek.com/sim-swap-scam-man-losing-life-savings-sparks-fbi-investigation-1706220.

[12] J. Murdock, “Newsweek,” 10 Feb 2021. [Online]. Available: https://www.newsweek.com/sim-swap-hackers-allegedly-stole-100m-cryptocurrency-celebrities-influencers-1568225.

[13] R. F. J. L. Y. L. S. N. Siqi MA, “An empirical study of SMS one-time password authentication in Android Apps,” Singapore Management University, 2019.

[14] C. L. C. R. A.-R. S. Alexandra Dmitrienko, “On the (In)Security of Mobile,” CASED, 2014.

[15] “Kaspersky,” 1 February 2019. [Online]. Available: https://www.kaspersky.com/blog/ss7-hacked/25529/.

[16] “Chaos Computer Club,“ 11 7 2024. [Online]. Available: https://www.ccc.de/en/updates/2024/2fa-sms.

[17] P. Paganini, “Security Affairs,“ 04 January 2024. [Online]. Available: https://securityaffairs.com/156920/hacking/orange-spain-ripe-account-hacked.html.

References

[18] P. Paganini, “Security Affairs,” 30 January 2024. [Online]. Available: https://securityaffairs.com/158329/cyber-crime/network-operators-credentials-found-in-dark-web.html.

[19] M. Dano, “Light Reading,” 10 March 2023. [Online]. Available: https://www.lightreading.com/security/verizon-at-t-t-mobile-and-dish-have-all-been-targets-of-hacks-this-year.

[20] S. Gatlan, “Bleeping Computer,” 4 January 2024. [Online]. Available: https://www.bleepingcomputer.com/news/security/russian-hackers-wiped-thousands-of-systems-in-kyivstar-attack/.

[21] Verizon, “Verizon business,” 2024. [Online]. Available: https://www.verizon.com/business/resources/reports/2024-dbir-data-breach-investigations-report.pdf.

[22] “EBA,” 2015. [Online]. Available: https://www.eba.europa.eu/regulation-and-policy/single-rulebook/interactive-single-rulebook/14575.

[23] “EBA,” 2022. [Online]. Available: https://www.eba.europa.eu/sites/default/files/document_library/Publications/Opinions/2022/Opinion%20od%20PSD2%20review%20%28EBA-Op-2022-06%29/1036016/EBA%27s%20response%20to%20the%20Call%20for%20advice%20on%20the%20review%20of%20PSD2.pdf.

[24] “Vodafone,” 29 June 2023. [Online]. Available: https://developer.vodafone.com/blog/introducing-sim-based-silent-identity-verification.

[25] M. B. F. H. D. N. O. R. D. M’Raihi, “RFC 4226 - HOTP: An HMAC-Based One-Time Password Algorithm,” 2005.

[26] S. M. M. P. J. R. D. M’Raihi, “RFC 6238 - TOTP: Time-Based One-Time Password Algorithm”.