magora

# How to Protect your App

## Authentication Model for Improving the Invulnerability of Your Software

Are you sure your mobile application is well protected? Is the business web platform or enterprise cloud system secure to the point of invulnerability? Do you still count on passwords as the sole authentication method or have you already moved to traditional 2FA?

This eBook will help you explore different ways to move to a more usable, cost-effective and secure model to protect your app or website.

# Terminology and Abbreviations

1 **Authentication.** Verification of unique information known or accessible to the person who is trying to access the data. The simplest example is entering an account password.

2 **One-Time Password (OTP).** Alphabetic or numeric code of 6-8 characters. This can be a code from an SMS or an application that generates codes.

3 **Two-Step Verification (2SV).** Login consists of two stages – for example, first you enter the password to the account and then the code from the SMS.

4 **Two Factor Authentication (2FA).** Login can also be spread across two stages, but the factors must be different. For example, you first enter a password and then enter a one-

# Authorisation Methods


## Single-factor

**Currently, password authorisation is the most common.** It is simple and familiar to most users. Passwords are used for authentication both within applications and on sites, in personal accounts and on user profiles.

The disadvantages of passwords, such as phishing, recruiting and others, are well known to everyone, but the simplicity and low cost of using this method allows it to take a strong lead among other methods.

**Single-factor authentication does not necessarily mean a password**. It can be biometric (for example, a fingerprint) or built on the basis of hardware identifiers (Touch Memory, USB token, smart card etc.). Such systems are largely devoid of the weaknesses of the password-only approach.

# Forms and registration

Two-factor authentication is usually a combination of a password and additional input medium: something that the user owns or a biometric parameter.


Here are the secondary factors that can complement or replace password protection:

- One-time SMS Codes
- One-time passwords generated on a smartphone (Google Authenticator, Nexus TruID, FreeOTP and others) or on a hardware generator (for example, Display Card)
- PKI (Public key infrastructure) certificates on a smart card, token or file
- Mobile apps that produce authorisation, or so-called mobile authentication
- Various types of biometric control procedures (fingerprint, retina scan, face scan, voice recognition etc.)

**What is the difference between two-step and two-factor authentication?**

Two-factor authentication consists of two steps, though the same formula is not always applicable to the two-factor method. The border between these concepts is tiny, to the point that often no distinction is made between them. For example, Twitter in a blog refers to its two-step authentication process as two-factor; Google equates these methods (however, the company offers both).

**An example of two-factor check:**

For remote access to the resources of my employer, I must pass two-factor authentication. The first factor is the account password, which I know. The second is the hardware token for generating OTP, which I have.

To log into the system on my behalf, an attacker must steal not only the password, but also the token issued to me, i.e. physically penetrate my apartment.

**An example of two-step verification:**

When you first log into Telegram from a new device, you receive a verification code on the phone – this is the first stage of authentication. For many users of the messenger, it is unique, but in the app's security settings you can enable the second stage – set your own password to be entered after the SMS code.

# Multi-Factor

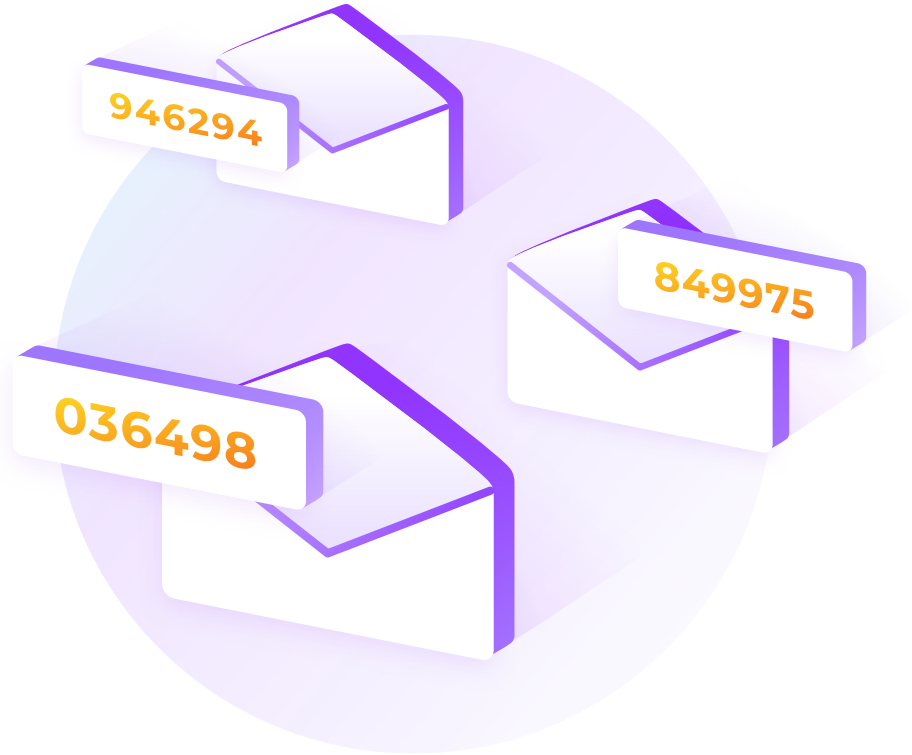
Multi-factor authentication uses factors of the following types: "I know", "I have", "I possess" (biometrics) and often refers to the two-factor model.

In fact, you can add as many factors as you want, combining those of similar types such as “I have” + “I posses”. One biometric factor can almost always be faked, but if you use several biometric factors for verification, it will be almost impossible to bypass such a system.

**For example**, you put your palm on the tablet, it reads and checks the pattern of the veins, the geometry of the palm and fingerprints. At the same time, using the built-in camera, the user’s facial geometry is also checked.

**The authentication system should be not only strong, but also convenient.**
You can implement a variety of different verification options, but if this is inconvenient, the user will either avoid using your system or will do everything to make their life easier, nullifying all your heaped-on security.

# Pros and Cons of Additional Factors




## SMS Disposable Codes

**SMS authentication is not an example of fully-fledged two-factor, but two-step.**
Due to the ease of use and widespread distribution of mobile phones, it is very common for the second factor to come via SMS. Many have come across this, for example, when authenticating on a website.

The method is very simple: After a successful login and password input, the verification system generates a random code that is sent to the phone number via SMS. The user enters the received code into the form on the website or in the app and successfully passes the authentication stage.

In reality, however, there are many inexpensive types of attacks that allow the interception of an SMS with a one-time code or simply the re-releasing of a subscriber's SIM card by another person.

> To protect users against reissuing of cards, some operators simply block the service of sending and receiving SMS for up to one day, and user authentication systems can detect a change in IMSI – the SIM card number – and not allow it to receive one-time codes before passing an additional stage.

## One-time passwords

**One-time password generators are divided into:**

Those that support open OATH standards

Those that use proprietary algorithms for generating one-time passwords

The one-time password generator can be software (usually an application on a smartphone, such as Nexus TruID) or hardware (various key rings or plastic cards with a display such as a Display Card). Hardware generators surpass their software counterparts in terms of security as they are isolated devices accessible only physically.

**According to the algorithms, OTP generators can be also divided into the following types:**

123 A one-time password based on the counter of the number of previously generated one-time passwords. Example – OATH HOTP.

A one-time password based on the time slot – that is, the counter here is the number of steps, usually 30 seconds each. For algorithms of this type, it is important to synchronise the time of the authentication server and the generator. Example – OATH TOTP.

Request-response – the system generates a code that the client transfers to the OTP generator (using buttons, scanning a QR code, listening to an audio recording), where a one-time password is generated in response. This one-time password is entered by the client for authentication on the website or in the program. An example is OATH OCRA.

At first glance, it seems more secure to generate a password based on time, since the password has a lifespan. In reality, however, time is not a secret parameter and the attacker actually already has data about this part of the process. In addition, the legal owner of the OTP generator doesn't recognise if the attacker has a duplicate generator. All authentication attempts — both user and intruder — will succeed.

If the counter password generation is used, a duplicate of the generator will be quickly detected, since the counter will be out of sync in the system and in the generator. It is also more difficult to guess the current position of the counter than to know the time.

The advantage of algorithms based on the request-response model is the possibility of mutual authentication of the server and the client, as well as the possibility of signing a transaction.

# PKI Authentication

This method is based on asymmetric cryptography. The client has a private key with which he or she signs an authentication server request. The system can verify the electronic signature using the client's public key (usually in the form of a certificate). This confirms the client's ownership of the private key.

A private or, alternatively, a user's secret key can be kept on a smart card, a cryptographic token, or simply in a file on a disk or removable drive. The token, like a smart card, basically uses a cryptographic chip with non-recoverable memory. The private key is generated directly inside the chip and this key cannot be copied to another medium. Keeping the secret key in a file is less secure, since someone can simply make a copy of the file.

**According to the algorithms, OTP generators can be also divided into the following types:**

A considerable advantage of certificate authentication is the practical impossibility of compromising a cryptographic token in the foreseeable time using a limited budget.

> If other verification methods require some secret information to be stored centrally on the server, then in the case of PKI authentication, the system doesn't store or operate anything with secret information. The only thing the server needs is to know which Certificate Authority (CA) issues the certificates. **It is simply not possible to leak any passwords or other secret parameters from the authentication server.**

## The Disadvantage of PKI

The low level of user convenience when working with tokens and cards for authentication, as well as the limitations of the devices to which you can connect a smart card or USB token, explain the rare distribution of this method.

# Mobile

Mobile authentication refers to the use of mobile devices, such as smartphones, to confirm entry.

A key pair is generated on the smartphone when registering with the system. The public key is sent to the authentication server, and the private key does not leave the smartphone and is used as in the usual certificate model.

**The process of checking this factor usually occurs as follows:**

1. The user enters username in the browser / application;
2. The server sends the request to the app on the smartphone. This is usually a random value (nounce);
3. The user confirms the verification on the smartphone by entering the PIN code;
4. The app produces a cryptographic signature and sends a signed response to the authentication server.

This is one of the most promising two-factor methods. On the one hand, it does not require the cost of purchasing additional devices. On the other hand, it provides a high level of protection based on public key cryptography. Moreover, mobile authentication is characterised by another advantage – the use of a separate logical channel.

> An example of very successful implementation of this authentication method is Nexus Personal Mobile – an application that works in conjunction with the Hybrid Access Gateway server. In addition to authentication itself, the application allows you to sign transactions. A transaction information message is displayed on the smartphone screen before the user authorises it.

# Biometric

There are currently many biometric authentication systems available. As biometric characteristics of a person they use:

- Fingerprint
- Facial geometry
- Retina
- Handwriting
- Heartbeat
- And others

> There are examples of centralised authentication systems by biometric parameters, but biometrics are mainly used for local authentication on the device: a laptop, a smartphone. Such systems allow you to exclude biometric data from the devices where they were collected and where they are checked.

Apps use these features to protect the input by replacing the PIN code. This does simplify user interaction.

**Remember:** Biometric authentication is based on the "similarity" of the current print, face, voice, etc. to the one that was studied by the system earlier. Small deviations from the sample present in the system will not lead to an authentication failure, in contrast to a small password change of even 1 character. That is, such a model cannot be called strict.

# FIDO U2F

The FIDO Alliance is addressing the problem of authentication on the Internet and proposes the standard U2F (Universal Second Factor) – a two-factor authentication protocol.

> This protocol allows the use of hardware cryptographic tokens but does not require complex centralised systems such as a single authentication server or public key infrastructure. This simplifies the application of the advantages of asymmetric cryptography and makes it accessible to a wide range of users and systems.

Among other things, the standard allows you to protect authentication processes from phishing and token cloning. Now such giants of the Internet industry as Google and Facebook allow its use.

# Which Model Should You Choose?

It is necessary to strive for the unification of authentication subsystems in various automated systems. Ideally, the system is single, but depending on the criticality of the system or the actions performed by the user an additional factor(s) may be added to the system.

# Criteria for Choosing Methods

When it comes to the right choice of methods and devices for two-factor authentication, in particular software, it's best to start with the analysis and prioritisation of the following criteria:

1. Security level (protection against unauthorised entry into information resources)
2. Infrastructure prerequisites (presence in the customer's corporate environment of authenticators for other access systems – for example, access control cards for access control systems – as well as domain certification services – for example, a certification centre from Microsoft)
3. Personnel categories (presence of remote workers in the company staff, for whom remote access to corporate resources is required)
4. Cost of ownership (includes not only software licences, but also the cost of additional equipment, such as USB readers, and the implementation and operation of the solution).

**In addition to the authentication factors themselves, pay attention to the security of the technology itself.**

Look for secure ways to transfer and store authentication data. Give preference to systems where use of the stored authentication data for registration in the system is not permitted and the intercepted data cannot be used for re-registration in the system.

# Conclusion

It is clear that enterprises should continue to evolve and improve their user authentication, moving beyond the limitations of passwords and traditional 2FA. Absolutely reliable systems do not exist. However, the multi-factor model can bring considerable benefits to corporate software security.

We highly recommend taking the following steps:

1. Add "something that I possess" to "something that I know" (login / password). This will provide you with a more robust authentication model.
2. Set various policies and ways of working by giving your employees a choice of options for their authentication mechanisms. If employees opt out of enhanced active modes, supplement your practices with passive modes.
3. Provide the ability to use corporate infrastructure credentials to log into applications, rather than creating new logins and passwords for the app in addition to corporate.
4. Include a variety of MFA options to address the needs of different users – for example, users with disabilities or those who are less tech savvy.
5. Use specialised systems to detect attempts to bypass (hack) the authentication system (such as SIEM class systems). Proper use of the Siem system can in many cases compensate for the shortcomings inherent in a particular authentication method.

Finally, it is never superfluous to assess the validity and security of your source code – contact the experts for consultation to make sure you're well-protected on all counts.

**If you have questions or need professional business consulting support, get in touch with our team and together we'll discuss all the issues in detail.**

**020 7183 5820**

**info@magora.co.uk**