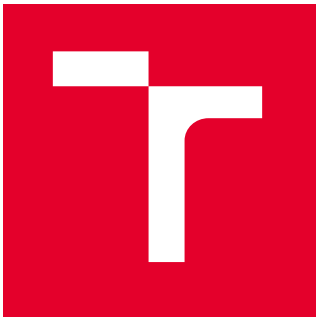


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

VÍCEFAKTOROVÁ AUTENTIZACE VYUŽÍVAJÍCÍ CHYTRÁ ZAŘÍZENÍ PRO PŘÍSTUP KE CLOUDOVÉ SLUŽBĚ

MULTI-FACTOR AUTHENTICATION USING SMART DEVICES TO ACCESS A CLOUD SERVICE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Luděk Huška**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Petr Dzurenda, Ph.D.**
SUPERVISOR

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Bc. Luděk Huška ***ID:*** 203704

Ročník: 2 ***Akademický rok:*** 2023/24

NÁZEV TÉMATU:

Vícefaktorová autentizace využívající chytrá zařízení pro přístup ke cloudové službě

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte problematiku současných metod vícefaktorového ověřování u moderních ICT systémů (např. bankovní služby, Microsoft služby atp.). Zaměřte se zejména na použité kryptografické protokoly, komunikační rozhraní, klíčový managment, a použité autentizační prvky. Vyhodnoťte možnost využití podpisů více stran (ang. Multi-party signatures) pro návrh autentizačního systému. Seznamte se s vývojem Android mobilních aplikací, platformou Nextcloud a systémem DECT. Po domluvě s vedoucím navrhněte a implementujte přístupový systém pro přístup do cloudové služby DECT. Systém bude využívat Android mobilní aplikaci a navržený koncept vícefaktorové autentizace uživatele pro přístup ke cloudovým službám.

DOPORUČENÁ LITERATURA:

[1] RICCI, S.; DZURENDA, P.; CASANOVA-MARQUÉS, R.; ČÍKA, P. Threshold Signature for Privacy-preserving Blockchain. In Business Process Management: Blockchain, Robotic Process Automation, and Central and Eastern Europe Forum. Münster, Germany: Springer, 2022. p. 1-15. ISBN: 978-3-031-16167-4.

[2] Android Developers [online]. Google [cit. 2022-09-01]. Dostupné z: https://developer.android.com/

Termín zadání: 5.2.2024 ***Termín odevzdání:*** 21.5.2024

Vedoucí práce: Ing. Petr Dzurenda, Ph.D.

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Tato diplomová práce řeší problematiku současných metod vícefaktorového ověřování u moderních ICT systémů. S rostoucími požadavky na bezpečnost a digitalizaci je ochrana dat klíčová. Práce poskytuje přehled o aktuálních technologiích a řešeních v oblasti vícefaktorové autentizace a zkoumá jejich integraci do systému pro přístup ke cloudové službě Nextcloud pomocí mobilní Android aplikace. Teoretická část se zabývá autorizací, autentizací, řízením přístupu a základy kryptografie. Praktická část zahrnuje vývoj Android aplikace a implementaci serverové části využívající protokol ECDSA. Výsledkem je funkční přihlašovací proces a aplikace pro Nextcloud, která zvyšuje bezpečnost dat a poskytuje uživatelům pohodlnější a bezpečnější způsob přihlašování.

KLÍČOVÁ SLOVA

Vícefaktorová autentizace, Android aplikace, Webová apliakce, Multi-party signatures, Nextcloud, Kryptografie veřejného klíče

ABSTRACT

This thesis addresses the issue of current multi-factor authentication methods for modern ICT systems. With increasing demands for security and digitalization, data protection is crucial. The thesis provides an overview of current technologies and solutions in the area of multi-factor authentication and explores their integration into a system for accessing the Nextcloud cloud service using an Android mobile application. The theoretical part deals with authorization, authentication, access control and cryptography basics. The practical part includes the development of an Android application and the implementation of the server part using the ECDSA protocol. The result is a functional login process and an app for Nextcloud that enhances data security and provides users with a more convenient and secure way to log in.

KEYWORDS

Multi-factor authentication, Android app, Web app, Multi-party signatures, Nextcloud, Public key cryptography

Vysázeno pomocí balíčku `thesis` verze 4.09; `https://latex.fekt.vut.cz/`

HUŠKA, Luděk. *Vícefaktorová autentizace využívající chytrá zařízení pro přístup ke cloudové službě*. Diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2024. Vedoucí práce: Ing. Petr Dzurenda, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Bc. Luděk Huška
VUT ID autora:	203704
Typ práce:	Diplomová práce
Akademický rok:	2023/24
Téma závěrečné práce:	Vícefaktorová autentizace využívající chytrá zařízení pro přístup ke cloudové službě

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Petr Dzurenda, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Také bych rád vyjádřil vděčnost své rodině za pomoc a podporu během celého mého studia. Konkrétně bych chtěl zmínit mého bratra a kamaráda Jakuba za konzultace a cenné rady.

Obsah

Seznam obrázků

Seznam výpisů

Úvod

V moderní době jsou kladeny čím dál větší požadavky na informační bezpečnost. Společně s růstem moderních technologií, digitalizace a zvyšováním závislosti na informačních systémech se bezpečnost dat stává klíčovou oblastí. Především se jedná o bezpečnost koncových uživatelů, kteří se stávají nejčastěji terčem útočníků. Tyto útoky mívají za cíl odhalit hesla uživatelů a to například za použití sociálních útoků, nebo útoků hrubou silou na dané heslo. Jejich premisa je získat, uhodnout a využít heslo pro přístup k utajovaným informacím.

Následky útoků bývají rozsáhlé a mnohdy napadené osobě způsobí i větší finanční ztrátu. Z tohoto důvodu se již některé společnosti snaží od hesel odprostit a vydat se cestou bezheslového přihlášení. Bezheslové přihlášení by mělo uživateli dovolit se přihlásit do služby bez potřeby si nějaké heslo pamatovat. Zároveň je však užívání mnohem pohodlnější a v celém procesu se zvyšuje obecná bezpečnost. V procesu se většinou využívá sekundární faktor pro ověření. Faktor může být například otisk prstu, sken obličeje, nebo nějaký fyzický hardwarový klíč. Sekundární faktor je pak využit místo hesla.

Standard dvoufaktorového ověření je již ukotven v nařízení SCA (Strong Customer Authentication). Toto nařízení určuje bankám v Evropské unii povinnost používat dvoufázové ověření. Je součástí druhé směrnice PSD2 (Payment Services Directive 2), která vstoupila v platnost 14. září 2019. Cílem SCA je zvýšit bezpečnost elektronických plateb a ochránit zákazníky před podvody. Dvoufázové ověření se stalo standardem nejen v bankovnictví, ale také v mnoha dalších oblastech, kde je ochrana citlivých údajů klíčová.

V této diplomové práci je proveden detailní přehled aktuálních informačních technologií a řešení v oblasti vícefaktorové autentizace. Práce zkoumá, jak tyto technologie mohou být integrovány do praktického systému pro přístup ke cloudové službě, s důrazem na bezpečnost a použitelnost. Výsledkem je návrh a implementace přístupového systému, který využívá vícefaktorovou autentizaci s pomocí mobilní aplikace, což zvyšuje bezpečnost a uživatelský komfort.

Práce je rozdělena do několika kapitol. První kapitola poskytuje teoretický úvod do problematiky autentizace, autorizace a kryptografie. Následující kapitoly se zabývají analýzou dostupných řešení na trhu a konkrétními implementacemi vícefaktorové autentizace. Závěrečná část práce se zaměřuje na praktický návrh systému pro vícefaktorovou autentizaci. Taktéž jsou zde popsány procesy přihlášení a registrace z pohledu uživatele.

1 Teoretický úvod

Tahle kapitola obecně popisuje základní teoretickou rovinu, jenž je potřebná k porozumění rozboru technologií řešení, která jsou nabízena na nynějším trhu a praktickému vyhotovení této práce. Především se jedná o základní pojmy jako autentizace, autorizace a řízení přístupu. Především v této kapitole bude kladen důraz na vícefaktorovou autentizaci.

1.1 Autentizace

Autentizace a autorizace [1] by se dala jednoduše popsat jako forma prokázání se a udělení validních oprávnění. Tyto procesy využívá každá osoba, která je nějakým způsobem konfrontována s prokazováním své identity a to jak v moderních informačních systémech, tak například při vchodu do budovy, kde je omezený přístup. V rámci této semestrální práce budou rozvinuty jen procesy, které mají spojitost s uživatelem, který se bude připojovat do webové aplikace.

Jak již bylo zmíněno autentizace je proces, který má za úkol ověřit identitu, tak aby mohli být uděleny určené oprávnění. V nejjednodušší formě je na tuto identifikaci v rámci webové služby nahlíženo jako na úlohu, kde uživatel zadá pár uživatelského jména a hesla. Vzhledem k tomu může být následně autorizován. V aktuálních systémech je na tento jednoduchý postup nahlíženo jako na zastaralý a zranitelný. Daná implementace umožňuje nespočet různých útoků, které mají za cíl odcizit účet a soukromá data oběti. Dokazují o tom i zprávy o incidentech [2] od Národního úřadu pro kybernetickou a informační bezpečnost, které jsou aktualizovány každý měsíc.
Z nich vyplývá, že útoky na kompromitaci účtů jsou druhé nejčastější po DOS (Denial of service) a DDOS (Distributed denial of service) útocích.

Tomuto problému se snaží předejít vícefaktorová autentizace, která není závislá jen na samotném heslu, ale taktéž na dalším údaji, který v rámci ověření poskytnut. To může potvrdit i statistika [3] společnosti Google z které vyplývá, že tato metoda autentizace pomůže ochránit ve většině případů.

Typy vícefaktorové autentizace

Ověřování pomocí této techniky [4] se dá obecně rozdělit do čtyř typů, které jsou využity jako druhý faktor při autentizaci:

- **Znalost** – Je závislá na informaci, kterou zná jen konkrétní osoba. To může být ověřeno bezpečnostní otázkou.

- **Vlastnictví** – Jedná se o fyzický předmět, kterým je možné se prokázat. Nejčastěji se jedná o čipové karty, USB (Universal serial bus) zařízení a další.
- **Charakteristika** – Je fyzický atribut jako otisk prstu, jedinečnost obličeje, hlasu a mnoho dalších.
- **Lokace** – Jako poslední možný faktor je lokace. A to z toho důvodu, že ne všechny služby musí být dostupné odkudkoliv. Je možné omezit použití jen na určité místo.

V návaznosti na zmíněné typy jsou níže vyjmenovány konkrétní způsoby [5], které jsou nejčastěji v praxi využity. Taktéž budou popsány jejich výhody a nevýhody.

- **SMS zpráva** – Tento typ autentizace požaduje po uživateli vlastnictví telefonního čísla a zařízení, na které je při přihlášení zaslán autentizační kód. Tímto kódem je možné se následně přihlásit. Tento typ více faktorového ověření je problematický hlavně z toho důvodu, že je pro útočníky jednoduché telefonní číslo vypátrat a s technikou SIM swapping ověření obejít. Přednost této techniky je hlavně její jednoduchost. Uživatel nemusí stahovat, či konfigurovat dodatečné nadstavby.
- **TOTP token** – TOTP (Time-based One-time Password) je šestimístný kód, který je dynamicky generován v externí aplikaci. Nejčastěji bývá platnost každého kódu jedna minuta, po které se kód změní. Tento kód je nutné odeslat během jeho platného časového horizontu. Tato metoda je považována za jednu z nejbezpečnějších. Její implementace však může být v určitých případech složitá a vyžaduje, aby si uživatel nainstaloval a nakonfiguroval aplikaci.
- **Emailový token** – Tyto tokeny jsou velmi podobné SMS (Short message service) tokenům, avšak jejich odeslání se realizuje za pomocí emailové služby. Tyto tokeny bývají častěji delší a složitější, než u SMS tokenů. Bezpečné použití tohoto procesu je závislé na bezpečnosti emailové schránky uživatele.
- **Hardwarový klíč** – Je token ve formě fyzického zařízení jako USB flash disk, nebo identifikační karta a podobně. I v tomhle případě se jedná o bezpečnou metodu. Jelikož pro útočníky je nemožné přes internet ukrást toto zařízení. Nevýhodou může být nutnost dalšího zařízení jako čtečka karty, nebo USB port v zařízení.
- **Biometrický údaj** – Validuje pomocí fyzické identifikace jako sken prstu, sítnice či obličeje. Tyto identifikace se považují za bezpečné, protože každý člověk má tyto fyzické rysy jedinečné. Při zjištění potenciální kompromitace se doporučuje stejnou metodu nepoužívat a zvolit jiný typ biometrického údaje

1.2 Autorizace a řízení přístupu

Autorizaci [6] si lze představit jako to, co je entitě povoleno dělat. Tento proces nastává po úspěšné autentizaci, která taktéž identifikuje uživatele a na základě této znalosti přiřadí práva, nebo oprávnění k provádění činnosti. V celém tomhle procesu je tedy důležité, kdo je autorizován, jaké oprávnění jsou mu přiřazeny, a kde je může uplatnit. Při autorizaci do reálného systému to znamená, že se určí povaha uživatele, jestli spadá do skupiny administrátorů, uživatelů nebo uživatelů se speciálními právy. Následně se mu přiřadí pověření číst, mazat, upravovat či měnit vlastníka souboru na základě tohoto faktu.

Řízení přístupu [7] je proces, který určuje, kdo může mít přístup k určitým datům nebo službám. Zahrnuje nastavení politik, procesů a nástrojů, které umožňují, blokují nebo omezují přístup jednotlivců nebo entit. Cílem řízení přístupu je zajistit, aby přístup k informacím byl poskytován pouze příslušným lidem. Zároveň ve vhodný čas a v souladu s předepsanými pravidly a předpisy. Tudíž rozdíl oproti autorizaci spočívá v tom, že autorizace se zaměřuje na konkrétní povolení, odepření individuální osoby k provádění konkrétních akcí nebo manipulaci s daty. Řízení přístupu oproti autorizaci se zabývá širším rámcem správy oprávnění a nastavení politik a procesů, které ovlivňují celkový přístup k informacím v organizaci.

Typy řízeni přístupu

Pro kontrolu přístupu [7] existují tři základní mechanismy. Tyto mechanismy jsou zmíněny a popsány níže:

- **DAC (Discretionary Access Control)** – V tomto modelu má vlastník, nebo tvůrce objektu kontrolu nad tím, kdo má k tomuto objektu přístup. Určuje oprávnění dané osoby a taktéž může udělit nebo odebrat přístup k objektům, nad kterými má kontrolu.
- **MAC (Mandatory Access Control)** – Zde jsou přístupová práva řízena centralizovaně a na základě předem definovaných pravidel a politik. Uživatelé zde oproti předchozímu případu nemůžou měnit nebo upravovat přístupová práva.
- **RBAC (Role Based Access Control)** – Jak už z názvu vypovídá jedná se o řízení přístupu na základě rolí. Každý uživatel má přiřazenou nějakou roli, ke které spadají příslušná oprávnění. Přístupové práva v tomto modelu jsou taktéž spravována centralizovaně.

Mimo tyto tři metody řízení přístupu, existuje mnoho dalších, které fungují na podobných principech s tím, že každý z nich může vykazovat různé charakteristiky. Například přístupové úrovně, šifrování a podobně.

1.3 Kryptografický základ

V této kapitole bude popsán kryptografický základ pro pochopení autentizačních mechanizmů a praktické implementaci diplomové práce. V podkapitolách bude probrána symetrická, asymetrická kryptografie, úvod do multi-party signatures a threshold signatures. V závěru bude provedeno vyhodnocení technologií založených na principu podpisu více stran k použití pro autentizaci uživatelů.

1.3.1 Symetrická kryptografie

V kryptosystémech existují dva základní systémy, a to symetrické a asymetrické, popsané v podkapitole 1.3.2. Jejich základní rozdíl [8] spočívá v práci s klíči pro šifrování a dešifrování. V symetrické kryptografii se pro šifrování a dešifrování používá stejný klíč, zatímco v asymetrické kryptografii se používají dva různé klíče pro šifrování a dešifrování.

Základní symetrické šifry

Mezi základní symetrické šifry [9] například patří:

- **DES (Data Encryption Standard)** – DES je historicky jeden z nejpoužívanějších symetrických algoritmů. Byl vyvinut v roce 1997 a uveden Národním úřadem pro normalizaci, nyní známým jako NIST (National Institute of Standards and Technology). Používá 64bitové bloky spolu s 56bitovým klíčem. Navzdory své tehdejší popularitě je dnes považován za zastaralý. Je to dáno především krátkou délkou klíče.
- **AES (Advanced Encryption Standard)** – Standard AES byl publikován institutem NIST v roce 2001. Používá klíče o délce 128 až 256 bitů, což zajišťuje vysokou úroveň bezpečnosti. AES je i v dnešní době používán a doporučován například úřadem NUKIB (Národní úřad pro kybernetickou a informační bezpečnost) v minimálních bezpečnostních standardech [10].
- **Blowfish** – Blowfish [11] je další známá symetrická šifra vyvinutá v roce 1997 Brucem Schneierem. Oproti jiným standardům není Blowfish patentován a má otevřený zdrojový kód. Je znám svou rychlostí a flexibilitou, protože umožňuje použití klíčů o délce od 32 do 448 bitů. V dnešní době se považuje za zastaralý.

Použití symetrické kryptografie v moderních systémech

Symetrická kryptografie je součástí a hraje klíčovou roli v mnoha moderních kryptosystémech. Je to dáno efektivitou a rychlostí šifrování a dešifrování. Použití symetrické kryptografie může být například při ukládání dat na disk, při šifrování

datového přenosu, v autentizačních protokolech, v mobilní komunikaci a v mnoha dalších oblastech.

1.3.2 Asymetrická kryptografie

Asymetrická kryptografie [12] používá dva různé klíče pro šifrování a dešifrování. Tyto klíče se nazývají veřejný a soukromý. Veřejný klíč může být volně distribuován ostatním osobám, zatímco soukromý klíč by měl být držen v tajnosti.

Šifrování a dešifrování vyžaduje oba klíče. Data šifrovaná veřejným klíčem lze dešifrovat pouze soukromým klíčem, a naopak. Tato metoda umožňuje vytvoření digitálních podpisů, které umožňují potvrzení platnosti zprávy a, že pochází od uvedeného zdroje. Této funkcionality bude využito v praktické části této práce.

Základní asymetrické šifry

Mezi základní asymetrické šifry patří:

- **RSA (Rivest, Shamir, Adleman)** – RSA [13] je jeden z prvních a nejpoužívanějších asymetrických algoritmů. Byl vytvořen v roce 1977 Ronem Rivestem, Adi Shamirem a Leonardem Adlemanem. Je založen na obtížnosti faktorizace velkých celých čísel. I v dnešní době je považován za bezpečný při použití delších šifrovacích klíčů. NUKIB uvádí minimální standardy pro použití šifrovací klíče v dokumentu o minimálních požadavcích na kryptografické algoritmy [10].
- **DSA (Digital Signature Algorithm)** – DSA [8] je standard pro digitální podpisy. Je nejčastěji používán k podepisování dokumentů online, čímž zajišťuje autenticitu a integritu podepsaných dat. Vzhledem k efektivitě a bezpečnosti je taktéž široce používán v různých aplikacích, od e-mailové komunikace po zabezpečení softwarové aktualizace.
- **ECDSA (Elliptic Curve Digital Signature Algorithm)** – ECDSA [14] je algoritmus založený na ECC (Elliptic Curve Cryptography). ECC [15] je moderní typ kryptografie, který nabízí vysokou úroveň bezpečnosti s kratšími klíči ve srovnání například s RSA.
  Algoritmus ECDSA je oblíbený především vzhledem k efektivitě využití zdrojů. Je používán často v zařízeních s omezenými prostředky jako mobilní telefony a podobně.
- **ECDSA (Elliptic Curve Digital Signature Algorithm)** – ECDSA [14] je algoritmus založený na ECC (Elliptic Curve Cryptography). ECC je moderní typ kryptografie, který poskytuje vysokou úroveň bezpečnosti s kratšími klíči ve srovnání například s RSA. Využívá eliptických křivek k vytváření digitálních

podpisů, což zajišťuje stejnou úroveň bezpečnosti jako tradiční algoritmy, ale s mnohem menšími klíči.

Tato efektivita je hlavním důvodem, proč je ECDSA oblíbený zejména v zařízeních s omezenými prostředky. To jsou například mobilní telefony, chytré karty a další. ECDSA se široce používá v různých aplikacích, od zabezpečené komunikace přes SSL/TLS až po blockchain technologie, kde je důležitá vysoká bezpečnost při minimálním využití výpočetních zdrojů.

Použití asymetrické kryptografie v moderních systémech

Jak z v poslední podsekce vyplývá asymetrická kryptografie je klíčovou technologií pro různé autentizační systémy a protokoly. Konkrétní použití v moderních systémech pak může být například:

- **Digitální podpisy** – Asymetrické klíče jsou základem pro digitální podpisy [16], které zajišťují autenticitu a integritu zpráv a dokumentů. Digitální podpis je jakási elektronická, šifrovaná známka, která ověřuje digitální informace. Potvrzuje, že zprávy pochází od konkrétní osoby a nebyly změněny. Tato vlastnost bude využita v praktické části práce, kde bude kontrolován podpis soukromého klíče, jestli opravdu náleží uživateli.
- **PKI (Public key infrastructure)** – PKI [17] jsou systémy a komponenty používané při zabezpečení internetové komunikace a transakcí. Má na starost správu a distribuci veřejných klíčů a certifikátů. Tyto certifikáty umožňují bezpečnou komunikaci a autentizaci na internetu.
- **SSL (Secure Sockets Layer)/TLS (Transport Layer Security)** – Hlavním účelem SSL a TLS [18] je zabezpečení komunikace mezi klientem a serverem. Protokoly můžou být taktéž využity pro zabezpečení e-mailu, VoIP (Voice over Internet Protocol) a jiné komunikace.

Asymetrická kryptografie hraje klíčovou roli v moderních autentizačních systémech a protokolech. Mimo zmíněné existuje mnoho dalších příkladů použití jako například šifrování dat, bezdrátová komunikace, zabezpečení cloudových služeb a mnoho dalších.

1.3.3 Multi-party threshold signatures

V této podkapitole bude probrán teoretický úvod do Multi-party a threshold signatures. Budou zde vysvětleny dané pojmy a k čemu se tento typ kryptografie využívá. Konec této podkapitoly bude věnován vyhodnocení možnosti použití podpisu více stran při autentizaci.

MPC (Multi-party Computation)

MPC [19] je typ kryptografie, který umožňuje několika stranám provádět společně bezpečné výpočty. Díky tomu je umožněno spolupracovat více stranám na provádění citlivých operací, jako je dešifrování a podepisování, aniž by odhalily své soukromé klíče. Tento přístup se využívá k ochraně soukromí a bezpečnosti dat při výpočtech, které by jinak vyžadovaly sdílení citlivých informací mezi stranami.

MPC se nejčastěji využívá v různých distribuovaných systémech, kryptoměnových peněženkách, blockchainových aplikacích a podobně. Například kryptoměnové peněženky [20] fungují tak, že technologie rozdělí soukromý klíč peněženky mezi několik stran. Tím je dosaženo zvýšení soukromí a snížení rizika, že se útočníci po zmocnění jednoho soukromého klíče dostanou k aktivům.

TC (Threshold Cryptography)

TC [19] používá techniku zvanou *sdílené tajemství*. Tajnou informaci rozdělí na více částí a předá je několika stranám. Aby bylo možné provést výpočet, musí se určitý počet těchto stran spojit a zkombinovat své části. Tento minimální počet stran, které se musí spojit, aby operace proběhla, se nazývá *práh*.

TC se používá k ochraně soukromí v cloud computingu, zabezpečení finančních transakcí a dalších citlivých aplikací, kde je zapojeno více stran.

Možnosti využití multiparty signatures pro návrh autentizačního systému

Multiparty signatures představují pokročilou kryptografickou techniku, která umožňuje více stranám podílet se na procesu vytváření jednoho digitálního podpisu. Díky této vlastnosti může tato metoda přinést do návrhu autentizačního systému další bezpečnostní prvky.

Možné využití MPC v autentizaci může zahrnovat například distribuovaný podpis pomocí více zařízení. Uživatel by mohl vlastnit více chytrých zařízení, která by byla spárována například přes Bluetooth. Místo otisku prstu by se pak ověřovalo, že uživatel vlastní daná zařízení. Tato zařízení by společně složila podpis a odeslala jej pro autentizaci.

2 Řešení dostupná na trhu

Tato kapitola je strukturována do tří hlavních částí. První podkapitola se zaměřuje na průzkum využívání vícefaktorové autentizace v moderních aplikacích. Budou zde probrány používané bezpečnostní metody, dostupná komunikační rozhraní, a také použité kryptografické metody a protokoly.

Druhá část se podrobně věnuje technologiím používaným pro autentizaci v těchto aplikacích. Budou zde vysvětleny klíčové procesy a způsoby, jakými tyto systémy fungují, s důrazem na jejich praktické nasazení a bezpečnostní aspekty.

Závěrečná část kapitoly se soustředí na cloudovou aplikaci NextCloud. Součástí bude vysvětlení NextCloudu a následně detailní srovnání notifikačních systémů, zhodnocení jejich výhod, nevýhod a omezení.

2.1 Aplikace implementující více faktorové přihlášení

V této sekci budou popsány praktické řešení bezheslového přihlášení a přihlášení do služeb, které využívají nástroje ze sekce 2.2. Důraz bude kladen na použité autentizační prvky, technologie, bezpečnostní opatření a jejich výhody, či nevýhody.

2.1.1 Česká Spořitelna George

Pro Českou Spořitelnu, jakožto banku, vzniká povinnost dle směrnice PSD2 (Payment Services Directive 2) [21] Evropské unie používat silnou zákaznickou autentizaci. To znamená, že proces autentizace musí zahrnovat alespoň dva ze tří možných faktorů: něco, co uživatel zná, jako heslo nebo PIN, něco, co uživatel má, jako mobilní telefon nebo kartu, a něco, co uživatel je, jako například sken obličeje nebo prstu.

Z tohoto důvodu používá Česká Spořitelna aplikaci George. George [22] je vytvořen jako bezpečnostní metoda, která potvrzuje totožnost uživatelů při přihlášení. Je navržen tak, aby dodržoval požadavky moderního digitálního bankovnictví, včetně přísných bezpečnostních standardů stanovených ve směrnici PSD2. Kromě samotné autentizace je bankovní aplikace navržena tak, aby umožňovala provádění převodů peněz, správu účtů, sledování transakcí a další služby.

Použité bezpečnostní metody

V mobilní aplikaci George je ověřována totožnost pomocí bankovní IDentity. Bankovní IDentita je způsob autentizace založený na unikátní kombinaci bezpečnostních metod. Tyto metody umožňují různé úrovně ověření totožnosti a zajišťují bezpečný

přístup k digitálním bankovním službám. Internetové a mobilní bankovnictví George odděluje bezpečnostní údaje do dvou základních skupin:

- **Heslo a SMS** – Při ověřování pomocí hesla a SMS je na telefon zaslán kontrolní kód, který uživatel zadá pro ověření totožnosti. Tato technika nevyžaduje, aby uživatel vlastnil chytrá zařízení, ale vystačí si se starším telefonem, který umí přijímat SMS zprávy.
- **George klíč** – K této metodě je nutné vlastnit chytré zařízení a mít na něm aktivovanou aplikaci George. Na telefonu se uživatel autentizuje při přihlášení a transakcích 6místným PINem, otiskem prstu nebo skenem obličeje v mobilní aplikaci George klíč.

Přihlášení z pohledu uživatele

Přihlášení podle oficiální dokumentace [23] probíhá tak, že se uživatel přihlašuje do webové aplikace standardně přes webové stránky České spořitelny. Na stránce zadá přihlašovací jméno, nebo klientské číslo. Při tomto pokusu o přihlášení přijde na mobilní zařízení uživatele zpráva s upozorněním na nutnost provedení potvrzení. Kliknutím na upozornění je otevřena aplikace George klíč s formulářem k přihlášení 2.1. Po kontrole o přihlášení a stisknutí tlačítka *potvrdit* je nutné zadat 6místný pin, který byl zvolen při aktivaci. Alternativy k tomuto pinu jsou otisk prstu, sken obličeje, nebo nově ověření hlasem. Aplikaci je následně možné zavřít.

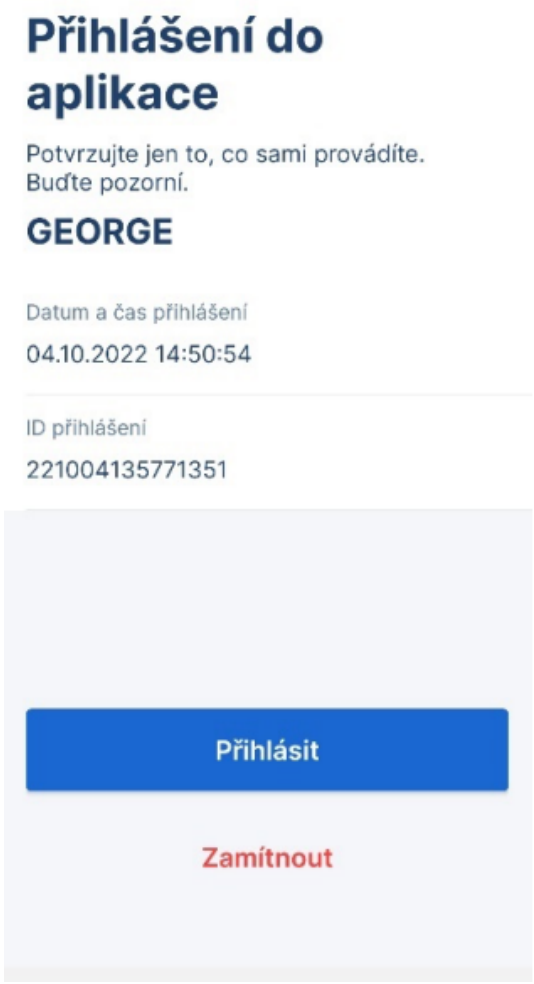
Použité kryptografické protokoly

Jak vyplývá z oficiální dokumentace [24], pro přihlášení uživatelů má Česká spořitelna vytvořené API (application programming interface), které využívá protokol OAuth2 blíže popsaný v sekci 2.2.1. Protokol umožňuje bezpečný přístup k uživatelským datům bez nutnosti sdílet přihlašovací údaje. Vzhledem k tomu lze předpokládat, že přihlašovací a ověřovací proces jsou v jejich aplikaci založeny právě na tomto protokolu. Avšak přesná funkcionalita aplikace, zacházení s klíči a použité protokoly pro implementaci přihlášení nejsou veřejně dostupné a patří k soukromému majetku České spořitelny. Komunikační rozhraní aplikace George zajišťuje bezpečný přenos dat mezi klientem a serverem pomocí šifrování HTTPS protokolu.

Závěr

Aplikace George od České spořitelny splňuje požadavky na silnou zákaznickou autentizaci dle směrnice PSD2. Využívá moderní bezpečnostní metody a kryptografické protokoly k zajištění bezpečnosti uživatelských dat. Předností tohoto řešení je jeho

snadnost a bezpečnost. Potenciální útočník by musel odcizit fyzické zařízení a obejít sekundární autentizaci, což je vysoce nepravděpodobné. Nevýhodou je nutnost vlastnit chytré zařízení, což může být pro některé uživatele problematické.



Obr. 2.1: Přihlášení do aplikace George. Převzato z [23].

2.1.2 Google přístupový klíč

Toto řešení [25] společnosti Google je moderní metoda pro vícefaktorovou autentizaci, která poskytuje vysokou úroveň bezpečnosti pro přístup k účtům Google. Tento systém eliminuje potřebu hesel a místo toho využívá biometrické údaje nebo zámek obrazovky zařízení k ověření totožnosti uživatele.

Výhodou oproti použití běžné autentizační aplikace je, že uživatel při přihlášení nemusí zadávat žádné přihlašovací údaje. Místo toho při přihlášení systém Google, prohlížeč nebo operační systém rozpozná, o jaký identifikační klíč se jedná. Pro

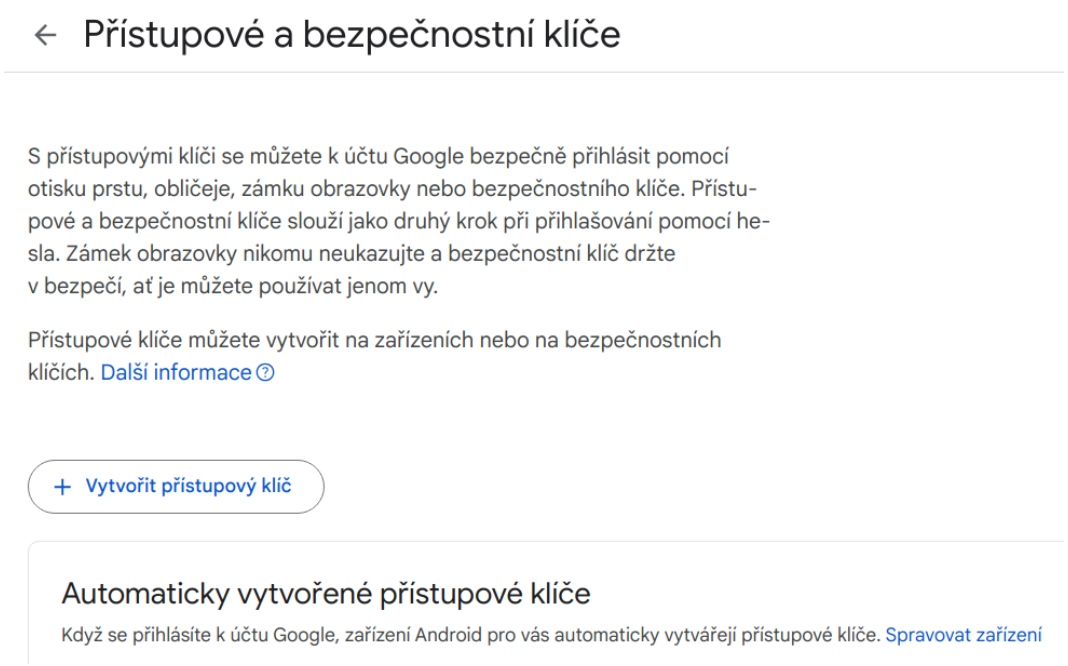
zajištění integrity je nutné se ověřit na daném zařízení pomocí hesla, otisku prstu nebo skenu obličeje. Navíc lze do Google Passkeys přidat více zařízení a vybrat si to nejvhodnější.

Použité bezpečnostní metody

Přihlášení pomocí Google Passkey je jednoduché a intuitivní. Skládá se jen z jednoho kroku. Uživatel se při pokusu o přihlášení do svého účtu Google nebo jiné služby, která podporuje passkey, ověří. Toto ověření může být realizováno pomocí jednoho z následujících faktorů: biometrie, PINu, vzoru na obrazovce nebo hardwarového klíče. Pro přihlášení na jiném zařízení může uživatel použít svůj telefon k ověření pomocí skenování QR kódu nebo podobného procesu.

Přihlášení z pohledu uživatele

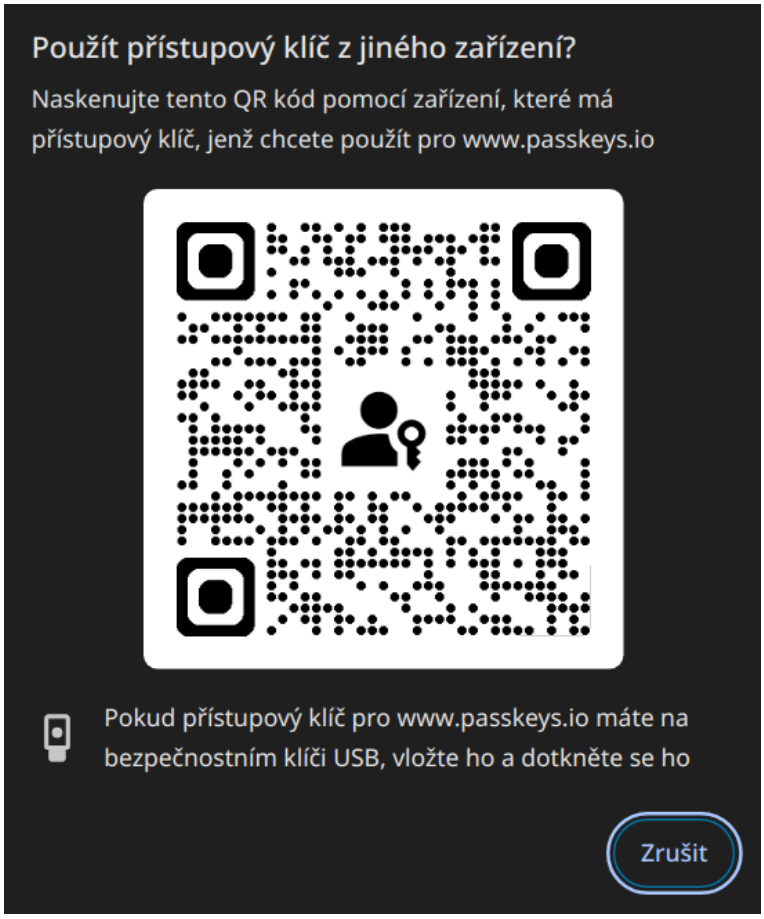
Pro používání přístupových klíčů si uživatel musí nejprve tuto službu zaregistrovat. Registrace může být náročnější ve srovnání s běžnými dvoufázovými ověřovacími metodami, a to především proto, že se jedná o novinku na trhu, která může obsahovat chyby a nepřesnosti. Pro registraci musí uživatel přejít na stránku `https://myaccount.google.com/signinoptions` a přihlásit se do svého účtu. Tato stránka je zobrazena na obrázku 2.2.



Obr. 2.2: Registrace google přístupového klíče.

Registrace je provedena kliknutím na tlačítko *Vytvořit přístupový klíč*, po kterém následuje interaktivní průvodce, který uživatele provede zbytkem procesu.

Po dokončení registrace je samotné přihlášení jednoduché. Po načtení služby Google v novém zařízení a výběru možnosti přihlášení je uživatel vyzván k naskenování QR kódu na zařízení, které obsahuje přístupový klíč. Tento QR kód je zobrazen na obrázku 2.3



Obr. 2.3: Přihlášení pomocí přístupového klíče na novém zařízení.

Pokud si prohlížeč nebo operační systém uživatele již pamatuje, automaticky odešle požadavek na autentizaci na v zařízení. V tomto zařízení se následně uživatel autentizuje pomocí vybrané metody ověření, a tím je automaticky přihlášen.

Použité kryptografické protokoly

Google přístupový klíč je založen na kryptografii veřejného klíče [26]. Dle oficiální vývojářské dokumentace se jedná o algoritmus ECDSA. Proces autentizace začíná

vytvořením klíčového páru během registrace. Veřejný klíč je následně odeslán a uložen na serverové straně, zatímco soukromý klíč zůstává bezpečně uložen na zařízení uživatele.

Během procesu přihlášení uživatel vytvoří digitální podpis, kterým podepíše výzvu svým soukromým klíčem. Tato podepsaná výzva je poté odeslána na server, kde se pomocí uloženého veřejného klíče ověřuje, zda byla výzva skutečně podepsána správným uživatelem.

Závěr

Google přístupový klíč představuje inovativní model vícefaktorové autentizace, který zcela eliminuje potřebu zadávat jakékoliv tradiční přihlašovací údaje, včetně hesel či uživatelských jmen. Místo toho autentizace probíhá prostřednictvím druhého faktoru. Tento přístup významně zjednodušuje proces přihlášení, zlepšuje uživatelský komfort a zároveň poskytuje vysokou úroveň zabezpečení, jelikož nevyžaduje zadávání žádných informací, které by mohly být snadno ukradeny nebo zneužity.

Bezpečnostní model Google přístupového klíče je mimořádně odolný proti útokům. Důvodem je, že soukromý klíč zůstává bezpečně uložený v zařízení uživatele a nikdy jej neopouští, což zásadně minimalizuje riziko jeho zneužití. Veřejný klíč, i když je sdílený a dostupný, nemůže být bez přístupu k soukromému klíči zneužit. Tento způsob ověření na principu asymetrické kryptografie s využitím algoritmu ECDSA, poskytuje vysokou úroveň ochrany a zabezpečení.

Jednou z nevýhod Google přístupových klíčů může být omezená kompatibilita s různými operačními systémy. Passkeys nemusí být podporovány mezi různými operačními systémy. To se může projevit zejména u uživatelů používajících například Linux, kde mohou nastat problémy s kompatibilitou. Kromě toho, použití Passkeys vyžaduje moderní zařízení vybavené technologiemi pro biometrické ověřování nebo hardwarové klíče. To může dostupnost této technologie pro některé uživatele.

2.1.3 Facebook bezpečnostní klíč

Toto řešení [27] společnosti Facebook je metoda vícefaktorového ověření využívající bezpečnostní klíče. Tyto klíče, obvykle malé hardwarové zařízení, umožňují uživatelům zabezpečit svůj účet. Bezpečnostní klíče často fungují v kombinaci s dalším faktorem, jako jsou SMS kódy nebo autentizační aplikace třetích stran, což zvyšuje úroveň zabezpečení.

Použití těchto hardwarových klíčů zvyšuje bezpečnost tím, že pro přihlášení vyžadují jejich fyzickou přítomnost. Tento přístup minimalizuje riziko phishingových útoků a dalších forem krádeží identity, protože útočník by musel mít fyzický přístup k hardwarovému klíči, aby se mohl přihlásit do účtu.

Použité bezpečnostní metody

Pro přihlášení pomocí této metody je nutné znát uživatelské jméno a heslo. Hardwarový klíč se používá jako další faktor k tomuto procesu. Facebook doporučuje [27] využívat tento způsob autentizace spolu s další metodou dvoufázového ověření, jako je kód v SMS zprávě nebo autentizační aplikace.

Mimo výše zmíněné Facebook zavádí další bezpečnostní metody, které doplňují celý tento proces. Například upozornění na nové přihlášení, které oznámí uživateli jakýkoliv pokus o přihlášení z nového zařízení nebo neznámé lokace. Další ochranné mechanismy jsou implementovány v rámci samotných klíčů, které mohou obsahovat například čtečku otisku prstu. Pokud je klíč odcizen a je detekován neoprávněný pokus o jeho použití, klíč se automaticky deaktivuje.

Přihlášení z pohledu uživatele

Pro používání tohoto typu autentizace je nejprve nutné zařízení registrovat v nastavení Facebooku. V nastavení se nachází sekce *Dvoufázové ověření*. Po jejím otevření si může uživatel vybrat mezi různými způsoby ověřování.

Obr. 2.4: Registrace bezpečnostního klíče.

Pro účely této podkapitoly je nutné zvolit možnost *Bezpečnostní klíče*. Okno registrace je zobrazeno na obrázku 2.4. Po kliknutí na tlačítko *Registrovat bezpečnostní klíč* se spustí interaktivní formulář, který uživatele provede celým procesem registrace.

Po registraci bezpečnostního klíče lze tento typ vícefaktorového ověření využívat. Pro přihlášení je nutné mít bezpečnostní klíč vložený v zařízení, na kterém se uživatel přihlašuje. V případě, že klíč obsahuje čtečku otisku prstu, je nutné provést také tuto formu autentizace. Tento proces může být doplněn ještě o ověření pomocí dalšího faktoru, jako časově omezeného hesla. Ověření pomocí časově omezeného hesla je blíže popsáno v kapitole 2.2.3.

Použité kryptografické protokoly

Autentizace pomocí hardwarového klíče je založena na kryptografii veřejného klíče. Na hardwarovém zařízení je bezpečně uložen soukromý klíč a při registraci se z tohoto zařízení nasdílí veřejný klíč. Nejčastěji se používají klíče ECDSA[29] nebo RSA [29]. ECDSA je v dnešní době preferovanější kvůli nižším hardwarovým nárokům.

Proces autentizace [30] začíná registrací klíče, kdy uživatel připojí hardwarový klíč k počítači nebo mobilnímu zařízení přes USB. Během tohoto procesu je generován pár klíčů - soukromý a veřejný. Soukromý klíč zůstává bezpečně uložen na hardwarovém klíči, zatímco veřejný klíč je registrován na serveru.

Když se uživatel pokouší přihlásit, server vygeneruje výzvu, náhodný řetězec znaků, a odešle ji na hardwarové zařízení. Hardwarové zařízení pak pomocí svého soukromého klíče podepíše tuto výzvu, čímž vytvoří kryptografický podpis. Tento podpis je jedinečný pro konkrétní relaci a uživatele. Podepsaná výzva je následně odeslána zpět na server, který ověří podpis pomocí veřejného klíče uloženého během registrace. Pokud podpis odpovídá, autentizace je úspěšná a uživatel je přihlášen. Proces může fungovat například na základě protokolu WebAuthn, který je blíže popsán v kapitole 2.2.2.

Závěr

Použití bezpečnostních klíčů jako metody vícefaktorového ověření přináší významné výhody i nevýhody. Mezi hlavní výhody patří vysoká úroveň zabezpečení. Bezpečnostní klíče a hardwarová zařízení snižují riziko phishingových útoků a útoků hrubou silou, protože pro úspěšnou autentizaci je nutná fyzická přítomnost klíče.

Nevýhodou tohoto řešení je potřeba vlastnit a mít vždy přítomný hardwarový klíč. To může způsobit nepohodlí, pokud uživatel klíč zapomene nebo ztratí. Další nevýhodou je finanční náklad na pořízení klíče, což může být pro některé uživatele

překážkou. Kromě toho, proces registrace a integrace klíče může vyžadovat technické znalosti, což může být náročné pro méně technicky zdatné uživatele.

2.1.4 Adobe Systems

V rámci tohoto řešení společnosti Adobe [31] je kladen důraz na přihlášení pomocí protokolu OAuth2. Tato moderní metoda vícefaktorové autentizace poskytuje uživatelům vysokou úroveň zabezpečení přístupu k jejich účtům a datům. Přihlášení probíhá prostřednictvím autentizace pomocí sekundární služby. To znamená, že uživatelé mohou využít své existující účty u jiných poskytovatelů, jako jsou Google, Facebook, Apple nebo Microsoft, pro přístup k službám Adobe.

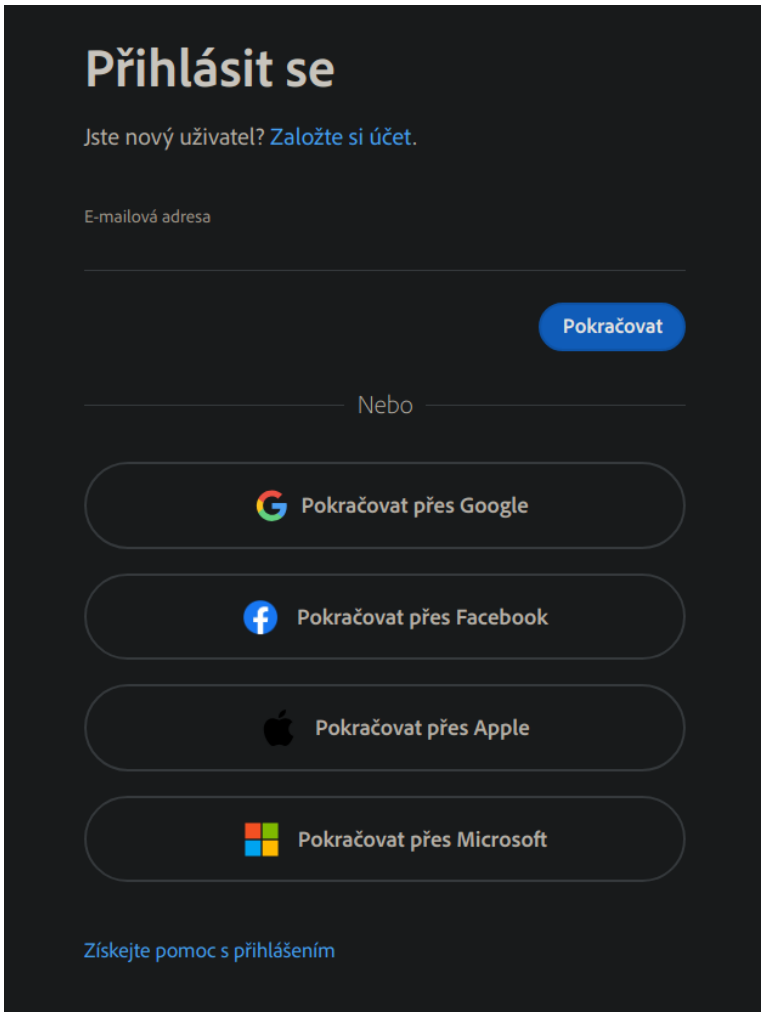
Použité bezpečnostní metody

Autentizační metody použité v tomto případě se odvíjejí od vybrané sekundární služby. Tyto služby, jako Google, Facebook, Apple nebo Microsoft, implementují vlastní bezpečnostní mechanismy autentizace. Může se jednat o metody, které byly zmíněny v předchozích sekcích, jako jsou Google Passkey (2.1.2) nebo Facebook bezpečnostní klíč (2.1.3).

Sekundární služby mohou používat i další metody autentizace, jako například Apple Face ID nebo Touch ID, zatímco Microsoft může využívat Windows Hello. Tento proces může být doplněn ověřením pomocí jednorázového hesla (2.2.3).

Přihlášení z pohledu uživatele

Uživatel se může přihlásit k této službě pomocí účtů registrovaných u třetích stran. Proces přihlášení začíná výběrem jedné z možností zobrazených na obrázku 2.5. Na stránce jsou k dispozici možnosti: *Pokračovat přes Google*, *Pokračovat přes Facebook*, *Pokračovat přes Apple* a *Pokračovat přes Microsoft*. Po výběru jedné z možností je uživatel přesměrován na autentizační stránku vybrané služby. V této fázi mohou nastat dva scénáře. Pokud je uživatel již autentizován, bude vyzván k povolení přístupu k jeho údajům pro Adobe. Pokud není aktuálně přihlášen, musí se přihlásit do vybrané služby. Po úspěšné autentizaci a povolení přístupu se uživatel vrátí zpět na původní stránku, kde je již přihlášen a může začít používat službu.



Obr. 2.5: Možnost přihlášení pomocí sekundární služby.

Použité kryptografické protokoly

Toto řešení [32]je založeno na protokolu OAuth 2.0, který je blíže popsaný v kapitole 2.2.1. Autorizační rámec OAuth 2.0 je protokol pro autorizaci, který umožňuje aplikacím získat přístup k uživatelským údajům bez potřeby sdílení hesel. Je využíván pro bezpečné autorizace mezi aplikacemi třetích stran a taktéž mezi poskytovateli služeb.

Závěr

Použití protokolu OAuth2 v autentizaci pro Adobe Systems přináší efektivní a uživatelsky přívětivou metodu přihlášení. Je toho docíleno pomocí sekundárních služeb jako Google, Facebook, Apple nebo Microsoft. Tento přístup umožňuje uživatelům využít stávající účty, což zjednodušuje proces přihlášení a eliminuje potřebu vytvářet

nové přihlašovací údaje. Nevýhodou je závislost na třetích stranách, což může ovlivnit dostupnost a bezpečnost při výpadcích jejich služeb. Další z nevýhod může být omezení kompatibility některých autentizačních metod na specifické zařízení nebo operační systémy. To může být pro některé uživatele zásadní překážkou. Celkově však OAuth2 poskytuje robustní řešení, které výrazně zvyšuje bezpečnost,komfort a usnadňuje přístup k účtům uživatelů.

2.1.5 Porovnání řešení

V této sekci budou porovnány jednotlivé řešení vícefaktorové autentizace, které byly probrány v rámci kapitoly 2.1. Tyto řešení budou hodnoceny na základě několika parametrů:

- **Snadnost použití** – Tento parametr hodnotí, jak snadné je pro koncového uživatele používat danou metodu autentizace. Zahrnuje intuitivnost procesu přihlášení, počet kroků potřebných k dokončení autentizace a požadavky na technické znalosti uživatele. Čím méně kroků a čím více intuitivní proces, tím lépe je metoda hodnocena.
- **Bezpečnost** – Tento parametr se zaměřuje na úroveň ochrany, kterou poskytuje daná autentizační metoda. Hodnotí, jak dobře metoda chrání proti různým typům útoků, jako jsou phishing, man-in-the-middle útoky, brute force útoky a další formy krádeže identity. Důležitým aspektem je také ochrana proti zneužití fyzického zařízení, pokud je použito.
- **Implementace** – Tento parametr hodnotí složitost a náklady spojené s implementací dané autentizační metody. Zahrnuje nároky na vývoj, nasazení, údržbu a kompatibilitu s existujícími systémy. Snazší implementace znamená nižší náklady a rychlejší nasazení, což je důležité pro organizace hledající efektivní řešení.

Z probraných řešení vyplývá, že nejsnadnější na použití je řešení od Adobe Systems, jelikož daná implementace neobsahuje přihlašovací údaje, ale pouze proklik na sekundární službu. Je však důležité vzít v úvahu, že snadnost použití se bude dále odvíjet od konkrétní sekundární služby. Alternativou k tomuto řešení může být přístupový klíč od společnosti Google. Tento typ autentizace vyžaduje chytré zařízení, na kterém není nutné mít žádnou externí aplikaci, což proces výrazně zjednodušuje. Naopak nejhorší z probraných možností je Facebook bezpečnostní klíč, který vyžaduje fyzické zařízení a přidává kroky v autentizaci navíc. Tyto kroky sice zvyšují zabezpečení, ale výrazně ovlivňují komfort uživatele.

Z hlediska bezpečnosti všechny zmíněné varianty poskytují vysokou úroveň zabezpečení proti běžným útokům, avšak Facebook bezpečnostní klíč vyniká díky využití pokročilé kryptografie a fyzických bezpečnostních prvků. Tyto prvky mohou

být navíc doplněny o další faktory v procesu přihlášení. Řešení České spořitelny George je rovněž velmi bezpečné díky kombinaci vícefaktorového ověření a důkladné implementaci bankovní identity.

Po stránce implementace lze vyvodit, že řešení od Google a Facebooku mohou být složitější na nasazení kvůli nutnosti integrace hardwarových klíčů a implementace protokolů, které spolupracují s daným operačním systémem. Naopak, řešení od Adobe Systems, které využívá OAuth2, je relativně jednoduché na implementaci, protože využívá existující autentizační infrastrukturu společností jako Google nebo Facebook.

Z průzkumu lze vyvodit, že neexistuje jednoznačně nejlepší řešení pro všechny scénáře. Avšak pro účely této práce, konkrétně přístup ke cloudové službě, může být nejlepší řešení založené na metodě kryptografie veřejného klíče. Tento přístup, podobně jako u České spořitelny, zajišťuje vysokou úroveň zabezpečení, spolehlivosti a uživatelského komfortu při autentizaci uživatelů.

2.2 Nástroje

V této kapitole budou zmíněny a popsány technologie, které se aktivně podílí na autentizaci do webových služeb. bude kladena hlavně na aktuálnost použitých nástrojů a co největší využití v dnešních službách. V této kapitole budou zmíněny a popsány nástroje, které se aktivně podílí na autentizaci do webových služeb. Pozornost bude kladena hlavně na aktuálnost použitých nástrojů a co největší využití v dnešních službách.

2.2.1 Oauth 2.0

Autorizační rámec OAuth 2.0 [33] je moderní autorizační protokol, který zjednodušuje proces autorizace tím, že umožňuje uživatelům poskytnout klientské aplikaci přístup k jejich datům uloženým na jiných internetových službách. Toho je dosaženo bez nutnosti sdílet s aplikací své přihlašovací údaje. Protokol poskytuje několik způsobů jakými může aplikace získat povolení k přístupu. TO může zahrnovat získání dočasného tokenu, který aplikace využívá pro autentizaci požadavků na server služby.

Definice OAuth rolí

OAuth definuje čtyři základní role, které se účastní procesu autorizačního tooku. Tyto role jsou:

- **Resource owner** – Je vlastník chráněného zdroje dat. Může přidělit, nebo odepřít přístup ke chráněnému zdroji.

- **Resource server** – Je to poskytovatel a hostitel chráněného zdroje. Může obsluhovat požadavky ke chráněnému zdroji.
- **Client** – Aplikace, která může přistupovat ke chráněnému zdroji na *resource server*u s oprávněními *resource owner*.
- **Authorization server** – Je to autorizační server, který dá uživateli přístupový token v případě jeho úspěšné autentizace.

Registrace klienta

Před použitím OAuth 2.0 musí klienti projít procesem registrace u autorizačního serveru. Tento server zajišťuje, že uživatelé jsou řádně ověřeni. Může obsahovat různé typy autentizace, jako jsou uživatelské jméno a heslo, biometrické ověření. Tento proces je nezbytný, protože uživatel musí být nejprve ověřen prostřednictvím autorizačního serveru, než mu bude umožněn přístup ke konkrétní službě nebo datům na cílovém serveru. Tím je zajištěno, že přístup je bezpečný a uživatelská data jsou chráněna.

Přihlášení uživatele

Pro přihlášení [34] je nejprve si definovat dva základní tokeny se kterými systém pracuje. Prvním z nich je *Access Token*, který je krátkodobý a umožňuje klientovi přístup k chráněným zdrojům serveru. Tento token má omezenou platnost, a po jejím uplynutí je nutné jej obnovit. Přístupový token je odesílán na resource server při každém požadavku na chráněný zdroj a jeho platnost je omezena dobou stanovenou autorizačním serverem.

Druhým typem tokenu je *Refresh Token*, který má delší platnost. Refresh tokeny jsou vydávány autorizačním serverem při prvním přihlášení a slouží k obnovení access tokenu. K obnovení dochází, pokud stávající access token vypršel nebo se stal neplatným. Refresh token se nikdy neodesílá na resource server, ale pouze na autorizační server, kde je ověřen a v případě platnosti je vydán nový access token.

Autorizační rámec OAuth 2.0 nabízí několik metod pro získání přístupového tokenu,. Každá z těchto metod je určena pro specifické typy aplikací a scénáře použití. Je to z důvodu zajištění flexibility protokolu pro různé typy klientů a zabezpečení služeb. Hlavními metodami pro získání přístupového tokenu jsou: *Authorization Code Grant*, *Implicit Grant*, *Resource Owner Password Credentials Grant* a *Client Credentials Grant*. Samotný proces přihlášení pak závisí na zvolené metodě pro získání autentizačního tokenu.

Tyto tokeny se přenášejí v čitelném formátu, což jejich držitelům umožňuje přístup k autorizovaným zdrojům na serveru. Z důvodu zajištění bezpečnosti je důležité, aby byly tokeny odesílány přes zabezpečené kanály, jako je HTTPS s implementací

TLS. Přenos tokenů přes nezabezpečený kanál by mohl vést k jejich kompromitaci a narušení služeb, ke kterým uživatelé přistupují.

Hodnocení

OAuth 2.0 poskytuje několik výhod [35], které zahrnují zejména uživatelský komfort. Díky tomuto protokolu uživatelé nemusí pamatovat na další přihlašovací údaje, což zjednodušuje proces přihlášení. Dalšími výhodami jsou škálovatelnost a flexibilita, které umožňují jeho nasazení v různých typech aplikací, ať už webových, mobilních nebo serverových.

Mezi nevýhody tohoto protokolu může patřit složitost implementace řešení. Ta může vyžadovat kroky navíc v plánování a provedení, aby byla zajištěna bezpečnost. Jako další výrazná nevýhoda může být závislost na autentizačních službách třetích stran. To představuje nevýhodu z toho důvodu, protože jakékoliv problémy nebo výpadky těchto služeb mohou nepříznivě ovlivnit funkčnost celé aplikace.

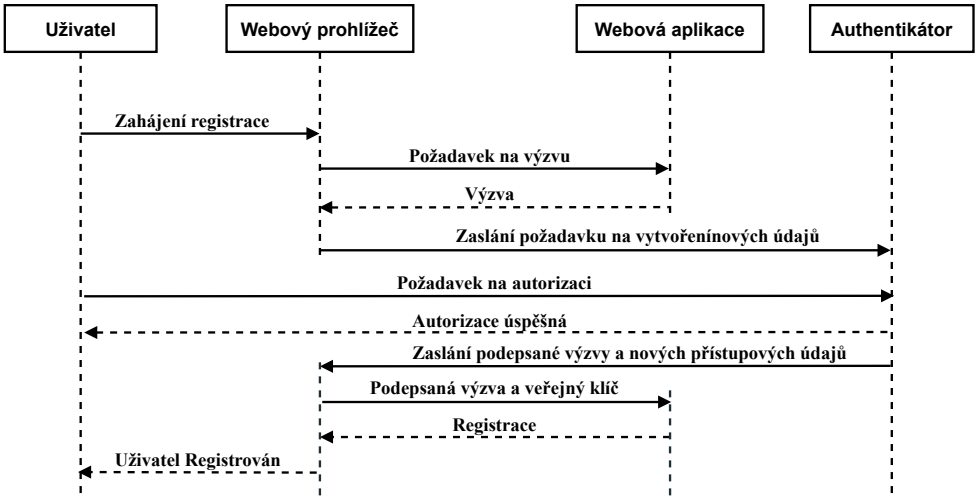
2.2.2 WebAuthn

Tento otevřený standard [36] umožňuje uživatelům bezpečné přihlášení bez nutnosti si pamatovat heslo. Funguje na konceptu, že se vygenerují dva kryptografické klíče. Veřejný klíč se odešle serveru aby mohl uživatele pomocí něj ověřit a soukromý si uchová samotné zařízení. Při přihlášení server zašle zařízení požadavek, které musí zařízení podepsat a server pak podpis ověřit. V samotném procesu figuruje taktéž ověření za pomocí druhého faktoru, kde je před podepsáním zprávy od serveru uživatel vyzván ať prokáže svou identitu.

Registrace

Celkový proces registrace [37] se skládá z několika částí. Tyto kroky budou rozkresleny na obrázku 2.7 a taktéž zde bude vysvětleno, jaký je jejich účel.

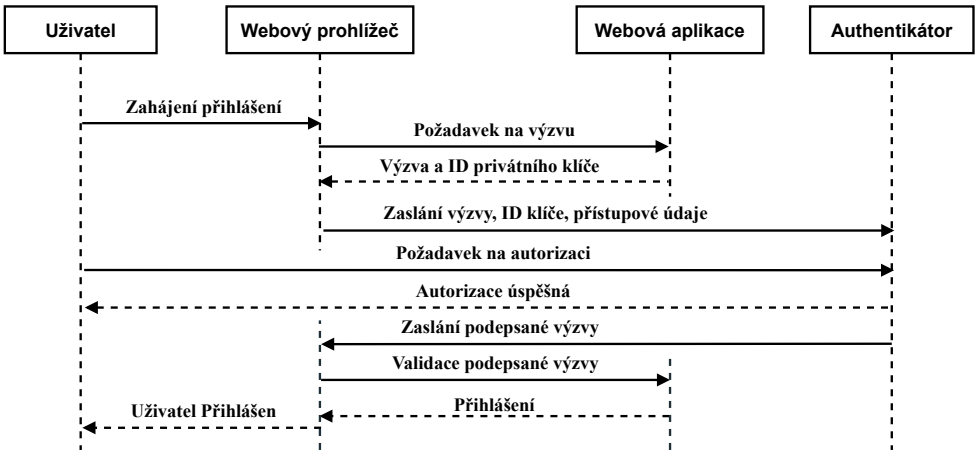
Uživatel spustí proces registrace kliknutím na příslušné tlačítko a tím vznikne požadavek na výzvu, kterou má zpracovat webový server. Tato výzva představuje data, která jsou zaslána klientovi. Po přijetí dat je následně odešle další výzvou na autentizační server po kterém požaduje nové přístupové údaje. Tento server po uživateli chce ať se autentizuje sekundárním faktorem. Po dokončení vytvoří server veřejný a privátní klíč. A pomocí privátního klíče podepíše obdržená data. Server následně odešle podepsané data, veřejný klíč a další související informace webovému serveru. Webový server ověří informace pomocí veřejného klíče jestli byly podepsány příslušným klíčem a tak uživatele úspěšně registruje.



Obr. 2.6: Proces registrace zařízení. Převzato z [37].

Přihlášení

Proces přihlášení [37] je podobný samotné registraci. Skládá se taktéž z osmi dílčích kroků. Tyto kroky budou rozkresleny a zpracovány jako v případě registrace.



Obr. 2.7: Proces přihlášení. Převzato z [37].

Uživatel spustí proces přihlášením pomocí příslušného tlačítka a zadáním svého uživatelského hesla. Ověřovací server následně odešle výzvu webovému prohlížeči a připojí k ní ID (Identification) soukromého klíče. Server v reakci odešle výzvu a ID soukromého klíče ověřovacímu zařízení. Toto zařízení ověří druhý faktor uži-

vatele. Požaduje po něm ověření jeho identity. Jakmile uživatel udělí oprávnění k ověření, ověřovací zařízení použije poskytnuté ID soukromého klíče k zobrazení vygenerovaného klíčového páru. Poté použije soukromý klíč k podepsání výzvy vydané ověřovacím serverem. Ověřovací zařízení odešle podepsanou výzvu a další informace pomocí výzvy webovému serveru. Tento webový server použije k ověření uložený veřejný klíč a uživatele autentizuje.

Hodnocení

Tento autentizační protokol splňuje všechny důležité požadavky, které jsou kladeny na bez heslovou autentizaci. Vysoká úroveň zabezpečení je doplněna o snadnost použití pro koncového uživatele. Implementace tohoto řešení se považuje za více náročnou než u jiných řešení. Taktéž může nastat situace kdy uživatel ztratí autentizační zařízení a tím si odstraní přístup do aplikace.

Toto přihlášení v praxi může vypadat tak, že se uživatel snaží registrovat, nebo přihlásit do nějaké služby. Místo standardních přihlašovacích údajů si však zvolí autentizaci pomocí jiné aplikace. Tímto uživatel udělí zvolené třetí straně oprávnění k přístupu k jeho účtu na ověřovacím serveru. Třetí strana pak může použít toto oprávnění k získání přístupu, který ji autorizuje k provádění akcí jménem uživatele. Vzhledem k tomu, že tento protokol slouží především k autorizaci a ne autentizaci tak zde nebude podrobně vysvětlen.

2.2.3 Časově omezené hesla na jedno použití

Časově omezené hesla na jedno použití (TOTP) je jedna z nejpoužívanějších metod dvou faktorové autentizace. Funguje na základě dvou předpokladů a to znalosti přihlašovacích údajů a znalosti dočasného hesla. Toto heslo mohou získat několika způsoby.

- **SMS zprávou** – Server odešle jednorázové heslo formou SMS zprávy na mobilní zařízení. Tyto hesla mají nejčastěji platnost několik málo minut.
- **Hardwarový bezpečnostní token** – Fyzické zařízení na jehož displeji se zobrazuje příslušný kód.
- **Mobilní aplikace** – Tato aplikace dynamicky generuje v určeném časovém horizontu přístupový kód. Generovaní přístupových kódů v aplikaci funguje na tom principu, že v zařízení je uložen tajný klíč. Tento klíč může ve spolupráci s aktuálním časem generovat heslo. K tomu je většinou využita některá z hašovacích funkcí. Tento proces je náročný z toho hlediska, že je důležité zajistit synchronizaci času mezi funkčním zařízením a serverem.

Hodnocení

Výhoda tohoto řešení je její bezpečnost. Sekundární faktor značně přispěje k zabezpečení přihlášení. Taktéž použití je velice jednoduché a intuitivní. Samotná synchronizace a implementace však může být problematická. Taktéž toto řešení neeliminuje používání hesel.

2.3 NextCloud a notifikační systém

V této podkapitole bude popsán NextCloud server, na kterém bude vytvořena praktická implementace diplomové práce. Jako druhý bod zde budou porovnány notifikační systémy a vybrán ideální systém pro tuto platformu.

NextCloud server

Platforma NextCloud [38] byla vybrána v zadání práce pro praktickou implementaci vícefaktorového ověření pro cloudovou službu. NextCloud server je otevřený, volně dostupný projekt, který slouží pro hostování souborů. Funguje na bázi klient-server a je přizpůsoben pro většinu operačních systémů. Vhodnost použití tohoto řešení již byla popsána v předchozí diplomové práci zabývající se tématikou přístupu ke cloudové službě pomocí čipových karet [39].

Projekt již využívá několik možností přihlášení. V základním nastavení je to standardní využití uživatelského jména a hesla. Pomocí aplikací lze tento systém rozšířit o další možnosti přihlášení. Může se jednat právě o přihlášení pomocí čipové karty, vícefaktorového ověření za pomocí e-mailu nebo například přihlášení pomocí jiné webové aplikace, jako je GitHub, Facebook a další. Tyto aplikace jsou dostupné z oficiálních webových stránek projektu [40].

Notifikační systém a komunikační model

V procesu autorizace pomocí mobilní aplikace budou hrát hlavní roli NextCloud server a chytré zařízení uživatele. V tomto procesu bude důležité zajistit, aby server mohl spolehlivě zasílat zprávy s požadavkem na autorizaci na konkrétní zařízení. Jelikož NextCloud server nemá nativně podporu žádné notifikační služby, je nutné najít a vybrat ideální řešení pro tuto úlohu. Na trhu existuje několik služeb, které se specializují na komunikaci mezi serverem a chytrými zařízeními uživatelů. Mezi nejznámější patří:

- **Firebase Cloud Messaging** – je služba poskytovaná společností Google [41]. Je to nejpoužívanější řešení pro zasílání notifikací a zpráv na mobilní a chytrá

zařízení. Jeho výhoda spočívá hlavně ve velmi snadné implementaci, široké podpoře platforem, spolehlivosti a optimalizaci.

- **Amazon Simple Notification Service** – je služba poskytovaná společností Amazon. Oproti Firebase Cloud Messaging je úzce spjata s dalšími službami Amazonu.
- **Pusher** – je, stejně jako Firebase Cloud Messaging, komunikační platforma, která umožňuje snadno implementovat zasílání zpráv na chytrá zařízení. Avšak z jejich porovnání [42] vyplývá, že Pusher nemá zaručené doručení zpráv a je fixován na jeden server, což může v určitých lokacích způsobovat delší odezvy a nedoručení zpráv.
- **OneSignal** – je poslední a druhá z nejpoužívanějších variant. Nevýhoda vůči ostatním je, že nenabízí žádnou bezplatnou verzi, tudíž v rámci této práce není použitelná.

V rámci tohoto projektu je Firebase Cloud Messaging ideální volbou. Nabízí snadné nasazení jak do webové aplikace, tak na mobilní zařízení. Má možnost využití na více operačních systémech, což z něj do budoucna dělá škálovatelné řešení. Další výhodou je jeho spolehlivost a rychlost, kterou zajišťuje robustní infrastruktura společnosti Google, minimalizující riziko ztráty dat a zaručující bezproblémové doručení zpráv. Dokumentace je rovněž dobře popsaná a obsahuje všechny potřebné informace k úspěšnému nasazení, což výrazně usnadňuje práci při vývoji aplikace.

3 Praktický návrh systému

Tato kapitola se zabývá praktickou realizací systému vícefaktorové autentizace pro cloudovou službu NextCloud. První v kapitole je podrobně popsána architektura systému a jak praktická část funguje. Dále se v kapitole nachází popis vytvoření a nastavení notifikačního systému. Na závěr se v kapitole nachází popis přihlášení a registrace. Obě tyto části jsou podrobně vysvětleny na komunikačních modelech, na kterých je ukázáno jak jednotlivé komponenty spolupracují. Závěr kapitoly je věnován grafickému návrhu aplikace.

3.1 Návrh architektury systému

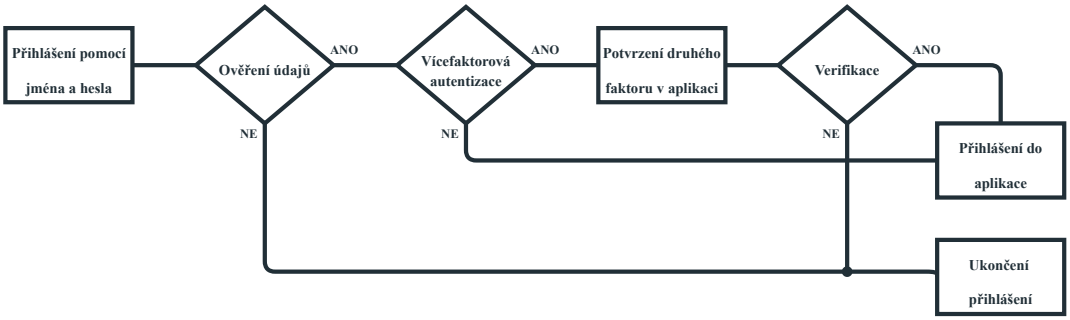
Pro pochopení návrhu je důležité si definovat dvě strany, které budou v procesu přihlášení figurovat. Na jedné straně to je server, na který se uživatel snaží autentizovat. Tento server bude popsán v sekci níže. Na straně druhé pak uživatel, který se pomocí vícefaktorové autentizace snaží na server přihlásit. Samotný princip této autentizace a jeho návrh pro praktické vyhotovení práce bude taktéž rozveden dále v této kapitole.

Využití vícefaktorové autentizace

V kapitole 2.1.5 bylo zjištěno, že nejefektivnější a nově populární metody přihlášení se všechny zakládají na principu bezheslové autentizace. Vzhledem k tomu je navržena architektura tohoto systému. Jelikož v NextCloudu není možné provést úplnou bezheslovou autentizaci, kde by uživatel zadal jméno a následně potvrdil druhý faktor. Což je dané vzhledem k charakteru tvoření aplikací na platformě NextCloud, obtížnou modifikací vnitřního kódu NextCloudu a samotnému nastavení projektu, na kterém aplikace funguje. Finální řešení tudíž obsahuje prvotní přihlášení, za pomocí jednoduše zapamatovatelného hesla jako například data narození uživatele. Následně je využita mobilní aplikace pro ověřování druhého faktoru. Díky tomu aplikace stále profituje z veškerých výhod dvoufaktorové autentizace.

Na obrázku 3.1 je zobrazen postup přihlášení uživatele, který buď používá, nebo nepoužívá, více faktorovou autentizaci. Lze vyčíst, že uživatel, který má tuto službu aktivovanou bude muset provést krok navíc. To sice proces přihlášení mírně zpomalí, ale zajistí výrazně vyšší bezpečnost. Z návrhu taktéž vyplývá, že použití vícefaktorové autentizace bude na každém z uživatelů a nebude povinně vyžadována.

Vzhledem k tomuto procesu je v rámci práce vytvořeno registrační nastavení pro uživatele. To umožňuje novým i stávajícím uživatelům libovolně aktivovat tuto



Obr. 3.1: Návrh postupu přihlášení.

službu a nakonfigurovat si ji pro svůj mobilní telefon. Proces nastavení služby společně s procesem přihlášení je vytvořen interaktivně, přehledně a jednoduše, aby odpovídal individuálním potřebám jednotlivých osob.

Komunikační model přihlášení

Pří úplném porozumění všem aspektům komunikačního procesu lze konstruovat modely pro autentizační mechanismy, jako jsou přihlašování a registrace. Model pro přihlášení, který představuje první z těchto implementací, je vizualizován na obrázku 3.2. Postup tohoto přihlášení probíhá následovně:

1) Uživatel přihlašuje do NextCloudu pomocí svého jména a hesla. Heslo při aktivovaném vícefaktorovém ověření může být například jeho datum narození. Následně jsou data odeslána na server.

2) Server po přijetí dat a jejich kontrole uživatele autentizuje pro potvrzení druhého faktoru. Je vygenerována výzva, která obsahuje klíč pro podepsání, uživatelovo identifikační číslo a identifikační klíč zařízení, na které se výzva odešle. Tyto údaje pak zašle pomocí HTTPS (Hypertext Transfer Protocol Secure) protokolu na API firebase serveru.

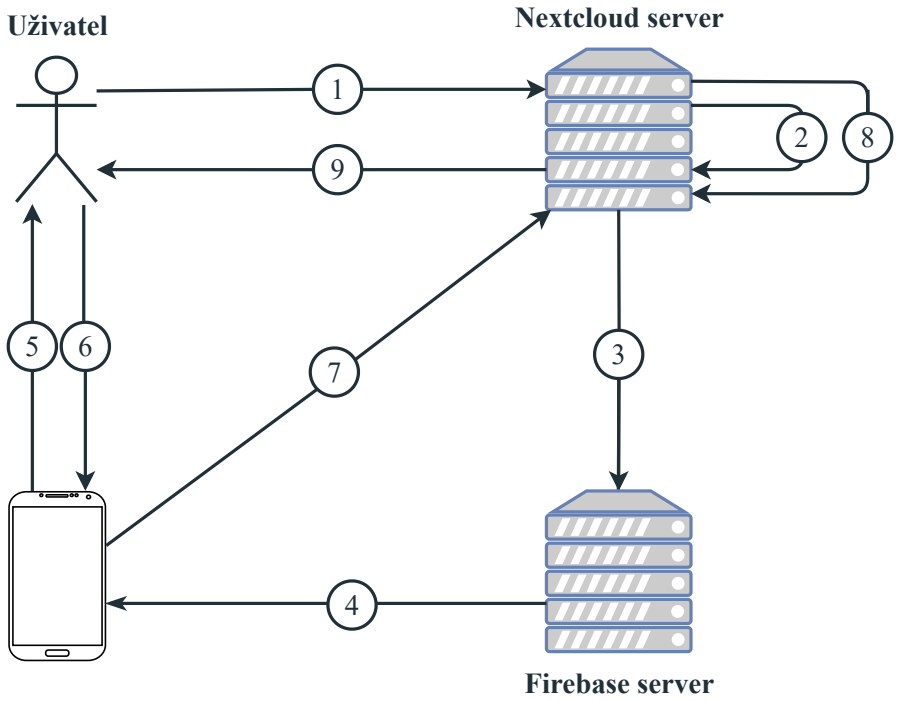
3) Vygenerované údaje jsou odeslány pomocí HTTPS protokolu na API firebase serveru.

4) Firebase server zpracuje požadavek na API a vyhledá si ve své databázi zařízení s požadovaným identifikačním číslem, na které dále odešle výzva přes HTTPS protokol.

5) Po přijetí výzvy na mobilním zařízení je vyvolána notifikace, která uživatele informuje o nutnosti potvrzení pro jeho přihlášení.

6) Uživatel může výzvu podepsat potvrzením druhého faktoru či tento požadavek zahodit a tím zabránit přihlášení. Prakticky to znamená, že po naskenování prstu může potvrdit příchozí výzvu. Ta je následně potvrzena soukromým

klíčem protokolu ECDSA, který je uložen jen na zařízení uživatele a je spjat jen s daným uživatelem.



Obr. 3.2: Komunikace na pozadí při přihlášení.

7) V tomto kroku je podepsaná výzva odeslána na API NextCloud serveru pomocí HTTPS protokolu.

8) Server má uložený veřejná klíč uživatele a díky němu může zkontrolovat pravost podpisu.

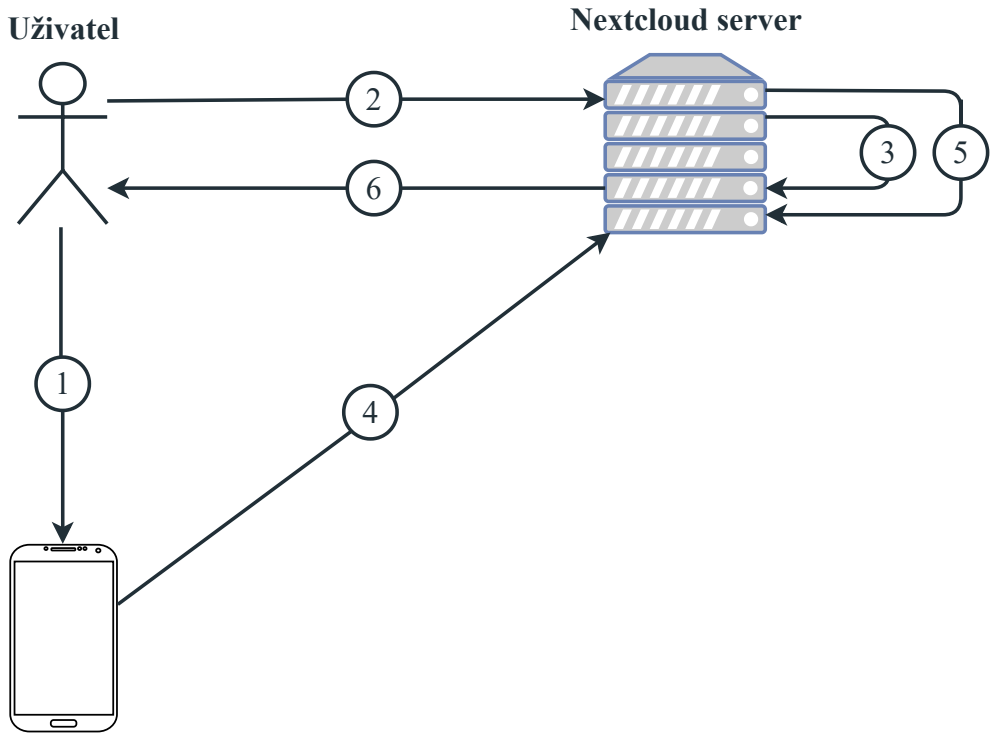
9) V případě správného podpisu je uživatel autentizován a autorizován.

Doplňující informace: V průběhu komunikace jsou všechny zprávy šifrovány pomocí protokolu HTTPS. Cílem je minimalizovat informace, které jsou sdíleny s Firebase serverem. Ten získá pouze informace o uživateli, který se autentizuje a o výzvě, která musí být podepsána soukromým klíčem. Zbytek komunikace během výměny podepsané výzvy probíhá pouze mezi uživatelovým chytrým zařízením a samotným serverem.

Komunikační model registrace

V komunikačním modelu registrace jsou do procesu zahrnuty jen tři strany. Uživatel, jeho chytré zažízení a NextCloud server. Postup registrace je vizualisován na obrázku

3.3. Při výměně informací jsou v tomto modelu taktéž všechny informace šifrovány pomocí protokolu HTTPS.



Obr. 3.3: Komunikace na pozadí při registraci.

1) Iniciální fáze zahrnuje stáhnutí a konfiguraci aplikace, která byla podrobně popsána v kapitole 3.3. Uživatel musí aplikaci udělit nezbytná oprávnění přístupu ke zdrojům, které aplikace vyžaduje. Tento krok vede k vytvoření FirebaseID, které je důležité pro další procesy a připravuje aplikaci pro proces registrace uživatele.

2) Následuje autentizace uživatele v rámci účtu, který bude používat vícefaktorovou autentizaci. Tento krok je klíčový z hlediska verifikace identity uživatele a zajištění, že další kroky budou prováděny oprávněnou osobou.

3) V dalším kroku na základě žádosti uživatele server generuje QR kód, který obsahuje uživatelské jméno a unikátní identifikátor účtu. Tento QR kód umožňuje rychlou a bezpečnou identifikaci a přihlášení. Podrobnosti o generování QR kódu a jeho specifikace jsou rozvedeny v kapitole 3.10.

4) Po načtení QR kódu pomocí aplikace dojde k aktivaci kryptografického procesu, během kterého je vygenerován klíčový pár algoritmem ECDSA. Soukromý klíč je uložen lokálně na zařízení uživatele za účelem zabezpečení, zatímco veřejný klíč je spolu s FirebaseID a dalšími relevantními informacemi odeslán na server NextCloud.

5) Server následně prověří přijaté údaje a na základě verifikace přiřadí k účtu FirebaseID a veřejný klíč. Tím je dokončena aktivace vícefaktorového ověření, které zvyšuje bezpečnost uživatelského účtu.

6) Konečná fáze zahrnuje informování uživatele o úspěšném dokončení registrace a aktivaci vícefaktorového ověření, čímž je proces autentizace uživatele završen.

3.2 Nastavení odesílání a přijetí notifikací

Implementace notifikační služby je zásadním krokem, který zajišťuje informování zařízení o pokusu uživatele o přihlášení. V rámci sekce 2.3 byl pro tento účel vybrán nástroj Firebase Cloud Messaging. Tento nástroj umožňuje především dynamicky odesílat notifikace na různé operační systémy a chytrá zařízení, ale taktéž využívat různé funkcionality navíc. Mezi ně patří například podpora mezi různými operačními systémy.

Registrace a vytvoření projektu

Prvním krokem je mít aktivní účet ve službě Google a následně se přihlásit do konzole Firebase na adrese `https://console.firebase.google.com`. Po úspěšném přihlášení je nutné vytvořit nový projekt podle instrukcí na hlavní stránce konzole. Po vytvoření a načtení projektu se na úvodní stránce objeví odkaz, který vás provede integrací s Android aplikací. Jelikož se zabýváme integrací notifikací ze serveru do aplikace, je nezbytné tuto integraci provést. To zahrnuje především vyplnění názvu android package name na hodnotu, která je pro tento projekt specifická: *com.example.mobiletwofactordect*. Po vyplnění je již možné stáhnout konfigurační soubory pro android studio. Tyto soubory je nutné v rámci mobilní aplikace importovat do složky *app*. Dále je nutné ve Firebase konzoli vytvořit integraci s Webovým serverem. Toho je dosaženo obdobným způsobem jak v předchozím případě. Tímto je prakticky integrace hotova a je možné již začít vytvářet odesílání a přijímání notifikací v rámci jednotlivých stran.

Odesílání a příjem notifikací

Pro odesílání zpráv je v projektu vytvořena třída *SendNotification*. Uvnitř této třídy se nachází funkce pro odeslání notifikace přes Firebase službu na zařízení uživatele.

V této funkci se nacházejí dva důležité prvky. První je integrace *mob.json* souboru, ve kterém se nachází informace, potřebné pro autentizaci a autorizaci při komunikaci s Firebase serverem. V další části kódu A.1 se odesílají data, o kterých musí

Výpis 3.1: Odeslání dat na Firebase server.

```
$mobData = json_decode(file_get_contents
    ('/var/www/html/custom_apps/twofactormobile/mob.json'), true);
```

být zařízení informováno, aby mohlo zajistit přihlášení pro konkrétního uživatele. Ve formátu JSON (JavaScript Object Notation) se proto odesílá v datech zprávy uživatelské jméno a výzva A.2, která je vygenerována na pozadí a pro ověření musí dojít k podpisu těchto dat soukromým klíčem uživatele.

Výpis 3.2: Data, která obsahuje výzva při odeslání.

```
"data":{
    "title": "Prosím autorizujte se v aplikaci",
    "body": "' . $userName . '",
    "challenge": "' . $challenge . '"
},
```

Tyto data jsou následně odeslány na API Firebase serveru. Firebase je schopný pomocí FirebaseID uživatele identifikovat zařízení, na které má notifikace dorazit a zprávu přepošle. V chytrém zařízení pak může dojít při příchozí notifikaci k dvěma variantám. K první dochází, když je aplikace zapnutá a automaticky se příchozí zpráva načte do paměti. V tu chvíli ji již lze automaticky potvrdit a není třeba další interakce. Druhá možnost nastane v případě, že je aplikace vypnutá a handler automaticky zachytí příchozí notifikaci. V tomto případě je nutné uživatelem příchozí zprávu otevřít a autentizovat se v aplikaci pomocí otisku prstu. Až následně jsou příchozí data načteny do paměti.

V opačném směru již nedochází k odeslání dat přes Firebase server. Mobilní telefon zasílá podepsané data na API NextCloud serveru, který je v aplikaci uživatelem nastaven. Toto opatření je z důvodu bezpečnosti, aby se nemohla další strana dostat k těmto datům.

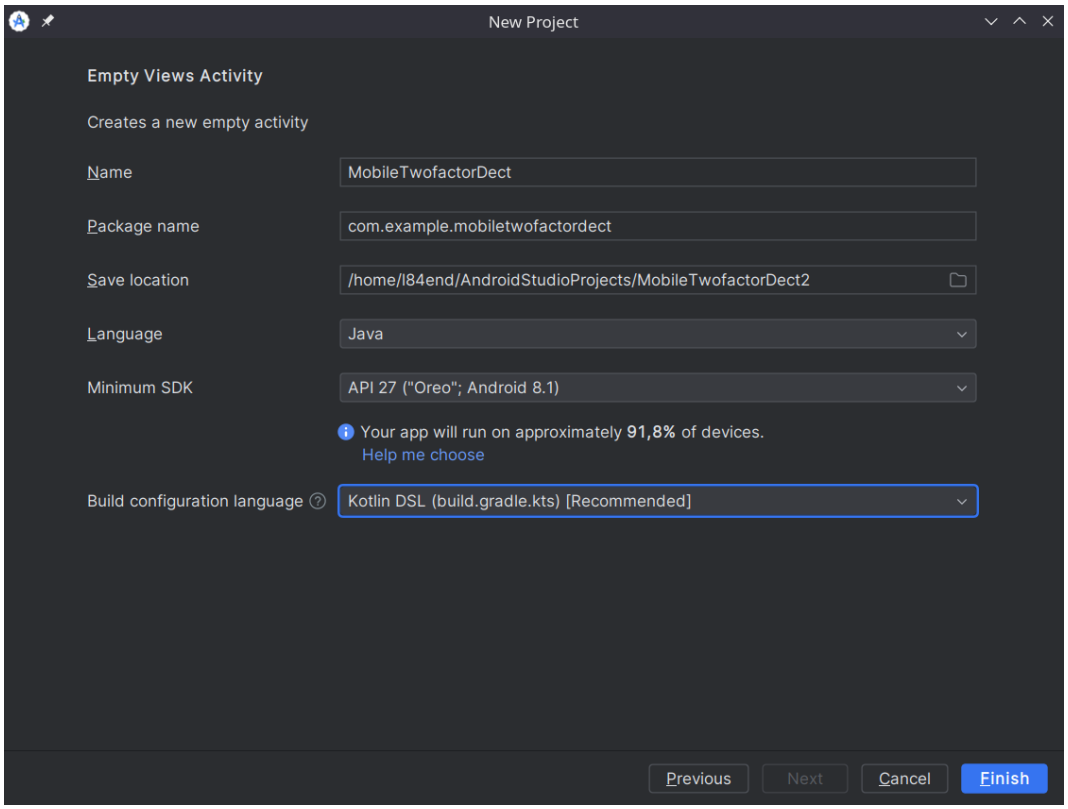
3.3 Návrh mobilní aplikace

Tato sekce se zaměřuje na technický návrh aplikace pro chytrá zařízení. Návrh zahrnuje komplexní přehled o implementaci a konfiguraci. Prvotní popis zahrnuje vytvoření projektu s detailní specifikací konfiguračních parametrů, jako je použitá verze SDK (Software Development Kit) a další relevantní nastavení.

Následuje část o oprávněních potřebných pro správnou funkčnost aplikace. Tato část podrobně rozebírá, která specifická oprávnění jsou vyžadována a jaký mají význam pro jednotlivé funkce aplikace. Dále je prezentován přehled použitých knihoven, včetně popisu jejich účelu a integrace do celkové struktury aplikace. Zvláštní pozornost je věnována kryptografické komponentě aplikace, kde je analyzován proces generování klíčů, výběr kryptografického algoritmu a metody ukládání klíčů.

Závěr kapitoly obsahuje návrh sekvenčních diagramů, které ilustrují procesy registrace a přihlašování uživatele, což pomáhá lépe pochopit tok dat a interakce mezi systémovými komponentami. Taktéž je zde popsán grafický návrh aplikace, kde jsou představeny klíčové vizuální elementy a uživatelské rozhraní, což poskytuje ucelený vizuální přehled funkčnosti a estetiky výsledné aplikace.

Vytvoření projektu



Obr. 3.4: Nastavení Android aplikace.

Projekt je vytvořen na základě prázdné šablony, s pečlivě zvolenými parametry, jak je znázorněno na obrázku 3.4. Je důležité si všimnout názvu projektu, který byl stanoven jako *MoobileTwofatorDect*, a automaticky vyplněného balíčku odvozeného od názvu projektu. Tyto informace jsou klíčové pro identifikaci umístění projektu,

jazyka, minimální verze SDK a jazyka pro konfiguraci sestavení. Zvláštní pozornost je třeba věnovat verzi SDK, neboť pro použité moduly a knihovny představuje minimální nutnou verzi.

Android povolení

Aby uživatel mohl provádět různé akce v aplikaci, tak je nutné přidělit patřičné povolení. Tyto oprávnění se definují v souboru *AndroidManifest.xml*, jenž se nachází ve složce projektu pojmenované jako *manifests*. Tyto oprávnění jsou znázorněny ve výpisu A.3. Každé z oprávnění má v aplikaci důležitou roli:

Výpis 3.3: Oprávnění v aplikaci.

```
uses-permission android:name="android.permission.INTERNET
uses-permission android:name="android.permission.
    ↪ ACCESS_NETWORK_STATE
uses-permission android:name="android.permission.CAMERA
uses-permission android:name="android.permission.USE_BIOMETRIC
uses-permission android:name="android.permission.USE_FINGERPRINT
uses-permission android:name="android.permission.FOREGROUND_SERVICE
```

- **INTERNET** – Umožňuje aplikaci připojit se k internetu, což je nezbytné pro přenos dat mezi klientem a serverem, stahování aktualizací nebo komunikaci se službami třetích stran.
- **ACCESS_NETWORK_STATE** – Umožňuje aplikaci přístup k informacím o stavu sítě, včetně toho, zda je zařízení připojeno k síti a jaký typ připojení je využíván. Tato informace je klíčová pro optimalizaci datových přenosů a správu síťových požadavků.
- **CAMERA** – Poskytuje aplikaci oprávnění k přístupu k fotoaparátu zařízení. Fotoaparát je využíván při procesu registrace, kde je potřeba pro naskenování QR kódu vygenerovaného na NextCloud serveru.
- **USE_BIOMETRIC** – Umožňuje aplikaci využívat biometrické senzory zařízení pro procesy autentizace. Při přihlašování do apliakce je nutné ověřit uživatelovu identitu, aby mohl potvrdit přihlášení jen oprávněný uživatel.
- **USE_FINGERPRINT** – Umožňuje aplikaci využívat čtečku otisků prstů na zařízení pro biometrickou autentizaci. Toto oprávnění je zásadní pro zabezpečené přihlašování a ověření při vstupu do aplikace.
- **FOREGROUND_SERVICE** – Toto oprávnění je třeba, aby aplikace mohla provozovat služby na pozadí. Notifikace o přihlášení může přijít i když uživatel nemá zrovna aplikaci spuštěnou.

Všechny tyto oprávnění a jejich podrobný popis lze najít na stránce [43].

Použité knihovny

V projektu je využito hned několik knihoven, které se starají o různou funkcionalitu. Tyto knihovny se přidávají do souboru *build.grandle.kts*, který se nachází ve složce *app*. V rámci projektu je snaha o použití co nejnovějších knihoven A.4, tak aby byly kompatibilní se zvolenou verzí androidu. Níže jsou popsány funkcionality knihoven:

Výpis 3.4: Externí knihovny použité v projektu.

```
implementation(platform("com.google.firebase:firebase-bom:32.7.4"))
implementation("com.google.firebase:firebase-messaging")
implementation("com.google.firebase:firebase-analytics")
implementation("com.squareup.okhttp3:okhttp:4.9.3")
implementation("com.google.zxing:core:3.4.1")
implementation("com.journeyapps:zxing-android-embedded:4.2.0")
implementation("com.google.android:play-services-vision:20.1.3")
implementation("org.bouncycastle:bcpkix-jdk15on:1.69")
implementation("androidx.biometric:biometric:1.1.0")
```

- **Firebase BOM** – Centralizuje správu verzí závislostí Firebase v projektu. Tento nástroj umožňuje jednoduchou a konzistentní aktualizaci Firebase modulů bez nutnosti specifikovat verze pro každý modul zvlášť, čímž značně zjednodušuje správu projektových závislostí.
- **Firebase Messaging** – Slouží k implementaci efektivního a spolehlivého mechanismu pro push notifikace a komunikaci mezi serverem a zařízeními uživatelů. Tato služba umožňuje zasílat notifikace do zařízení uživatele v reálném čase.
- **Firebase Analytics** – Poskytuje rozsáhlé nástroje pro sběr dat o interakcích uživatelů s aplikací. Data získaná pomocí tohoto nástroje pomáhají analyzovat efektivitu uživatelského rozhraní a chování.
- **OkHttp** – HTTP klient poskytující vysokou efektivitu a spolehlivost pro síťovou komunikaci v aplikacích. OkHttp podporuje automatické opakování požadavků, sledování připojení, cachování a kompresi, což zlepšuje výkon a robustnost síťových operací.
- **ZXing Core** – Knihovna pro generování a dekódování různých formátů čárových kódů a QR kódů. ZXing je široce používán pro integraci funkčností souvisejících s čárovými kódy v mobilních aplikacích, což zahrnuje skenování a tvorbu QR kódů.
- **Google Play Services Vision** – Obsahuje API pro pokročilé vizuální funkce, jako je rozpoznávání obličejů, textů, čárových kódů a dalších vizuálních prvků v aplikacích.

- **Bouncy Castle** – Obsahuje rozsáhlou sadu kryptografických algoritmů a nástrojů pro Java a C#. Bouncy Castle poskytuje funkce pro šifrování, dešifrování, digitální podpisy, vytváření a ověřování certifikátů a mnoho dalších kryptografických operací.
- **Biometric** – Poskytuje nástroje pro integrování biometrických autentizačních metod, jako je otisk prstu nebo rozpoznávání obličeje do aplikace.

Mezi klíčové knihovny v projektu, které mají zásadní vliv na bezpečnost a funkcionalitu aplikace, patří *Biometric*, *Bouncy Castle* a *OkHttp*. Knihovna *Biometric* zajišťuje implementaci vícefaktorové autentizace s využitím biometrické metody otisku prstu. *Bouncy Castle*, která umožňuje bezpečné kryptografické operace včetně generování a ověřování digitálních podpisů s použitím algoritmu ECDSA. A *OkHttp* hraje klíčovou roli v bezpečné síťové komunikaci, protože podporuje HTTPS, což zaručuje šifrovaný a bezpečný přenos dat mezi klientem a serverem. Tato kombinace knihoven představuje spolehlivý základ pro ochranu uživatelů a jejich dat.

Kryptografie

V rámci práce je implementována knihovna *Bouncy Castle*, která podporuje širokou paletu kryptografických operací, včetně protokolu ECDSA. Protokol ECDSA je zvolen především kvůli jeho efektivnosti ve srovnání s RSA, jak je diskutováno ve studii [44]. Hlavní výhody ECDSA zahrnují menší nároky na velikost klíče při ekvivalentním zabezpečení a rychlejší generování podpisu, což je ideální pro zařízení s omezenými výpočetními kapacitami.

V práci je vytvořena praktická implementace ECDSA a bezpečnému zacházení s klíči. Nejdůležitější funkci lze vidět ve výpisu A.5. Důraz je kladen na proces generování klíčů, který využívá funkce *KeyPairGenerator.getInstance* k inicializaci generátoru klíčů s algoritmem pro eliptické křivky. Klíče jsou následně ukládány do bezpečného úložiště Android KeyStore [45], aby byly chráněny proti potenciálním útokům a neoprávněné extrakci. Toto úložiště je navrženo tak, aby zajistilo maximální bezpečnost uložených klíčů a minimalizovalo možnost jejich zneužití. Do aplikace jsou rovněž implementovány další důležité funkce pro správu klíčů. Mezi nejdůležitější patří *signMessage* ve které probíhá podepisování zpráv veřejným klíčem od konkrétního uživatele. K veřejným klíčům v Android KeyStore lze přistupovat pomocí *getKey*. Mezi další funkce pak patří *deleteEntry* pomocí které lze z Android KeyStore odstranit záznamy klíčů.

Výpis 3.5: Generování klíčů pomocí ECDSA.

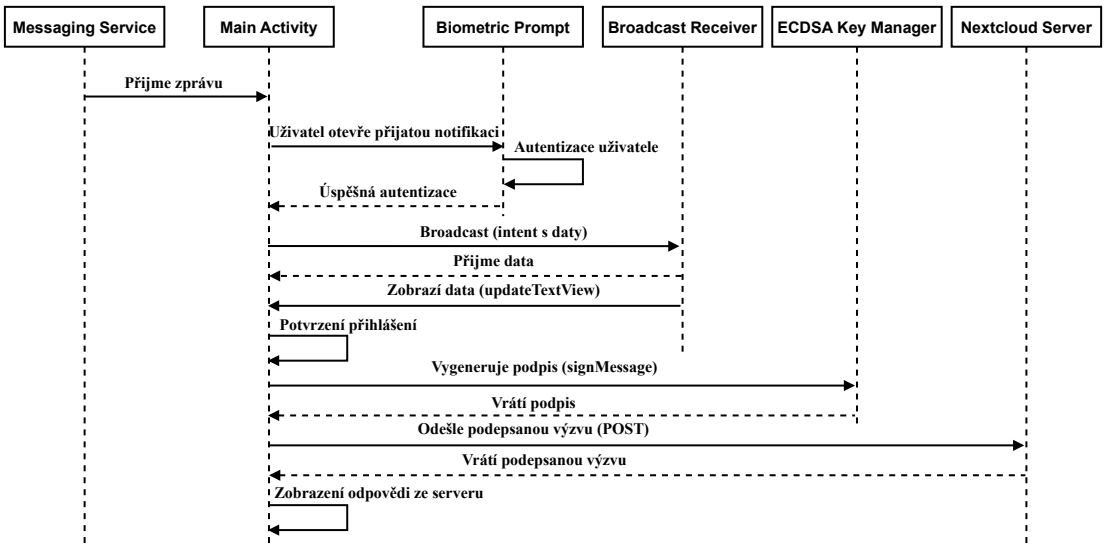
```
public void generateKeyPair(String alias) {
    try {
        KeyPairGenerator kpg = KeyPairGenerator.getInstance(
            KeyProperties.KEY_ALGORITHM_EC, "AndroidKeyStore");
        kpg.initialize(new KeyGenParameterSpec.Builder(
            alias,
            KeyProperties.PURPOSE_SIGN | KeyProperties.
                ↪ PURPOSE_VERIFY)
            .setDigests(KeyProperties.DIGEST_SHA256,
                    KeyProperties.DIGEST_SHA512)
            .build());

        KeyPair kp = kpg.generateKeyPair();
    } catch (NoSuchAlgorithmException | NoSuchProviderException |
            InvalidAlgorithmParameterException e) {
        e.printStackTrace();
    }
}
```

Sekvenční návrh operací v aplikaci

Na obrázku 3.5 je zobrazen sekvenční diagram návrhu pro přihlášení. Tento diagram popisuje jednotlivé interakce a kroky, které se odehrávají v rámci přihlašování uživatele a zpracování příchozí notifikace. Průběh procesu je následující:

1. **Příjem zprávy:** Nejdříve *MyFirebaseMessagingService* přijme zprávu, která obsahuje titulek, tělo zprávy a výzvu.
2. **Biometrická autentizace:** Pro pokračování v procesu je nutné, aby se uživatel autentizoval. Autentizace probíhá pomocí biometrických údajů, konkrétně otisku prstu. Tento krok je zajištěn komponentou *BiometricPrompt*.
3. **Předání dat:** Po úspěšné autentizaci se data z přijaté zprávy předávají do hlavního uživatelského rozhraní aplikace pomocí *BroadcastReceiver*. Díky tomu budou data dostupná pro další práci.
4. **Zobrazení dat:** *MainActivity* pomocí metody *updateTextView* zobrazí přijatá data uživateli. Uživateli se tímto zobrazí přesné informace o tom pod kterým uživatelem se snaží přihlásit a toto příhlášení následně potvrdit, či zamítnout.
5. **Generování podpisu:** Po kliknutí na tlačítko pro přihlášení *MainActivity* spustí proces generování podpisu pomocí *ECDSAKeyManager*.
6. **Odeslání dat na server:** Vygenerovaný podpis je následně odeslán na server NextCloud pomocí POST požadavku.

7. **Odpověď ze serveru:** Server zpracuje přijatý požadavek a vrátí odpověď zpět do aplikace. Uživatel je tímto informován o výsledku přihlášení.

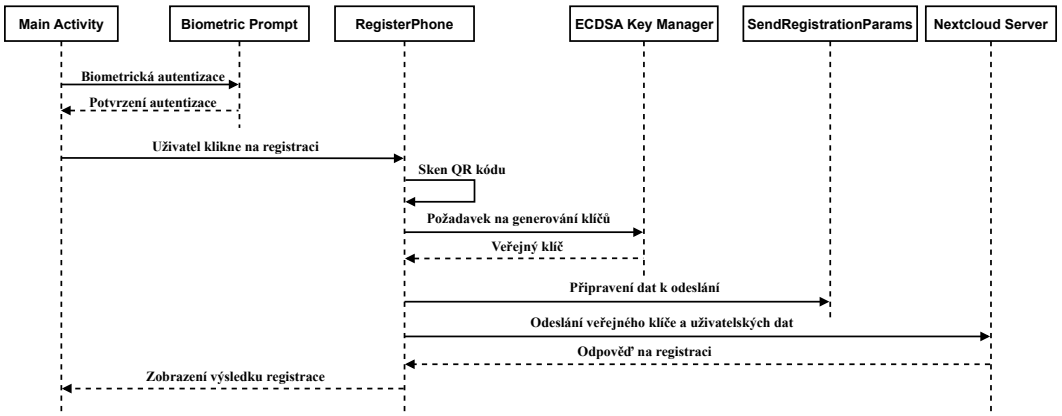


Obr. 3.5: Sekvenční model přihlášení na straně aplikace.

Pro upřesnění je na obrázku 3.6 zobrazen sekvenční návrh pro registraci. Tento diagram zobrazuje jednotlivé dílčí kroky stejně, jak v případě přihlášení. Diagram zobrazuje fungování jen na pozadí aplikace pro chytrá zařízení. Průběh procesu je následující:

1. **Spustění Aplikace:** Uživatel spustí aplikaci.
2. **Biometrická Autentizace:** Ihned po spuštění aplikace je zavolána komponenta *BiometricPrompt* pro biometrickou autentizaci.
3. **Přesměrování na registraci:** Uživatel klikne na možnost registrace v nabídce menu na hlavní stránce aplikace.
4. **Sken QR kódu:** V této sekci se aktivuje skenování QR kódu za pomocí třídy *RegisterPhone*.
5. **Extrakce dat:** Data jsou extrahovaná z QR kódu a jsou zpracována, včetně uživatelského jména a tajného kódu. Tento tajný kód musí být odeslán v registraci, aby nemohlo dojít k podvržení cizím účastníkem.
6. **Generování klíčů:** Třída *ECDSAKeyManager* generuje pár klíčů pro uživatele. Soukromý klíč uloží bezpečně do Android KeyStore pod jménem uživatele.
7. **Nastavení dat k odeslání:** *SendRegistrationParams* připraví JSON objekt s daty pro registraci, včetně veřejného klíče.
8. **Odeslání dat na NextCloud server:** Data jsou odeslána na server pomocí HTTPS požadavku s příslušnými detaily na NextCloud server.

9. **Odpověď ze serveru:** Server zpracuje přijatý požadavek a vrátí odpověď zpět do aplikace. Uživatel je tímto informován o výsledku registrace.

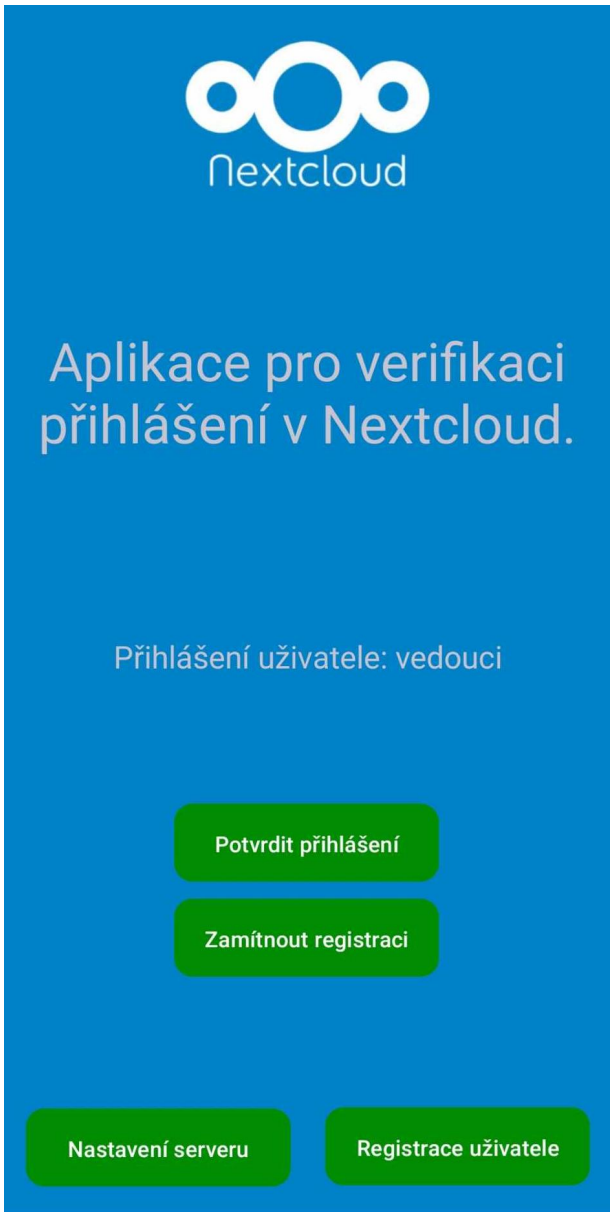


Obr. 3.6: Sekvenční model registrace na straně aplikace.

Grafický návrh aplikace

Jak lze na obrázku 3.7 zpozorovat, tak aplikace je zacílena na přehlednost a snadnou navigaci. Rozvržení je postaveno na ConstraintLayout, který umožňuje flexibilní rozvržení prvků podle rozměrů obrazovky. To zajišťuje kompatibilitu s většinou chytrých zařízení a rozlišeními jejich displejů. Barevné schéma aplikace je inspirováno originálními barvami NextCloudu. Toto barevné schéma je obohaceno o výrazně světlou zelenou barvu, která zdůrazňuje interaktivní prvky. U těchto prvků byla cíleně zvolena takto výrazná barva, aby bylo jasné, že s nimi může uživatel nějakým způsobem manipulovat. Aplikace je taktéž na horní straně obrazovky doplněna NextCloud logem.

Na hlavní stránce se standardně nacházejí dvě tlačítka *Nastavení serveru*, které přesměruje uživatele na stránku, kde může ručně zadat na jaké internetové adrese se NextCloud nachází. Druhé tlačítko *Registrace uživatele*, které přesměruje na stránku s čtečkou QR kódu, kterou uživatel potřebuje při registraci. Tlačítka uprostřed pro potvrzení a zamítnutí přihlášení se zobrazí jen v případě příchozí notifikace. Po odkliknutí jedné z možností se znovu skryjí na pozadí.



Obr. 3.7: Hlavní obrazovka mobilní aplikace při příchozím požadavku.

3.4 Návrh NextCloud aplikace

V této části je popsán praktický návrh aplikace na straně NextCloud serveru. Jako první je v této kapitole popsáno založení aplikace a použité knihovny v projektu. Dále se pokračuje návrhem registrace vícefaktorové autentizace, popisem vytvořených API, stavovým diagramem jak aplikace funguje a nakonec grafickým návrhem aplikace.

Založení aplikace

Aplikace pro NextCloud jde automaticky generovat v jejich systému `https://apps.NextCloud.com/developer/apps/generate`. Je zde třeba vyplnit několik údajů vzhledem k povaze projektu.

- **Název aplikace** – TwofactorMobile
- **Celé jméno autora** – Luděk Huška
- **E-mailová adresa autora** – xhuska02@vutbr.cz
- **Domovská stránka autora** – github.com/xhuska02/twofactor_mobile
- **Kategorie** – Zabezpečení
- **Souhrn** – Vícefaktorová autentizace za použití chytrého zařízení.
- **Popis** – Vícefaktorová autentizace za použití chytrého zařízení.

Po vyplnění se automaticky založí a stáhne projekt, který lze do NextCloudu importovat a začít jej vyvíjet. Podrobné informace k vývoji jsou dostupné z oficiálních vývojářských stránek NextCloud.

Použité knihovny

Oproti Android aplikaci jsou v tomto projektu doinstalovány jen dvě externí knihovny. Tyto knihovny jsou definované v souboru *composer.json*, který se nachází v hlavní složce projektu. Jedná se o knihovny *google-auth-library-php* a *qr-code*.

První výše zmíněná knihovna má v kontextu aplikace na starost získávání autorizačního tokenu, který je potřebný k odeslání notifikací přes službu Firebase Cloud Messaging. Tato funkcionalita je implementována ve třídě *SendNotification.php*. Dále má za úkol nastavení HTTP hlaviček, vytvoření a odeslání POST požadavku na Firebase s příslušnými daty a návrat odpovědi.

Druhá z knihoven *qr-code* je implementována pro generování QR kódu, který je využit k registraci. Výhoda této knihovny je rychlost a možnost využití různých formátů jako PNG, SVG, EPS nebo binární formát.

WebAuth autentizační protokol

Tento moderní autentizační protokol [36] je určený k zabezpečení přístupu k online službám. Využívá kryptografické bezpečné autentizační metody. Umožňuje uživatelům přihlašování k různým webovým službám bez použití hesla například za pomocí rozpoznání otisku prstu, obličeje, bezpečnostních klíčů a podobně.

V rámci předešlé práce [39] je již v projektu tento autentizační protokol implementován. A je možné rozšířit projekt o metodu navrženou v rámci této práce.

Vytvořené API

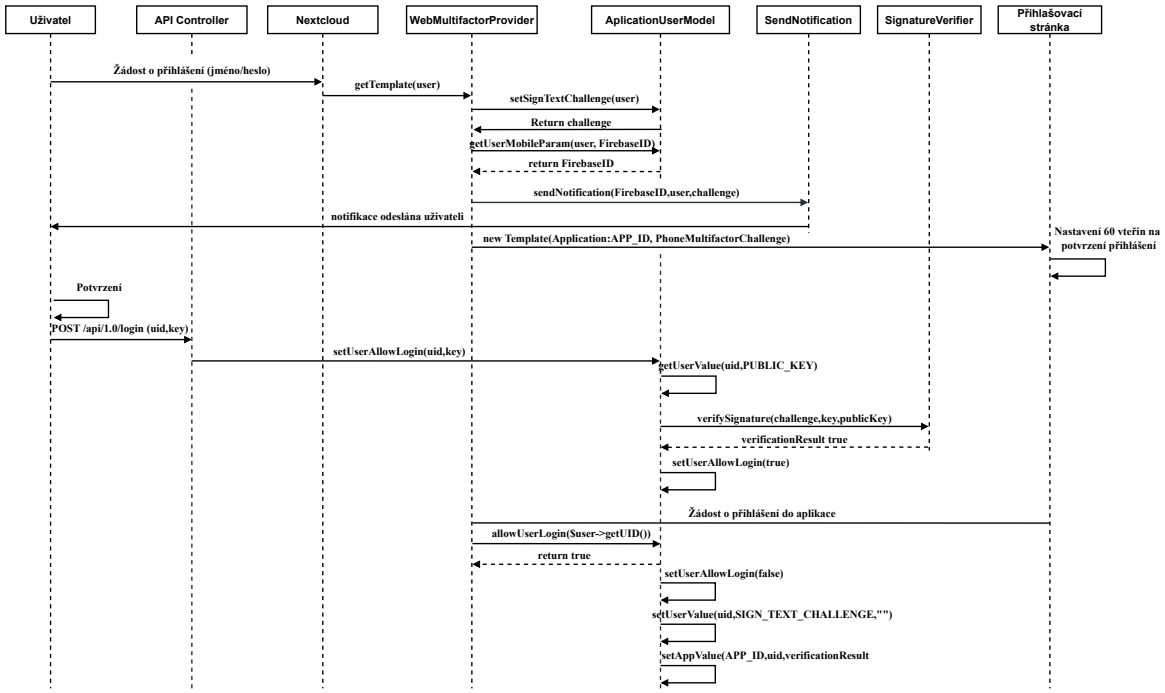
API endpointy jsou v NextCloud aplikaci definovány v souboru *routes.php*, který se nachází ve složce *appinfo*. Ty mají za úkol mapovat HTTP požadavky na metody ve třídách kontrolérů, které pak zpracovávají jednotlivé operace. Detaily vytvořených API endpointů jsou vypsány níže:

- **page#index** – Slouží uvnitř aplikace pro přesměrování na stránku s registrací.
- **mobile_api#checkLogin** – Na tomto endpointu se ověřuje příchod autentizačního tokenu od uživatele. Pokud je token přítomen a platný, umožňuje pokračovat v procesu přihlášení.
- **mobile_api#login** – Tento endpoint umožňuje přijímat a zpracovávat požadavky na uživatelské přihlášení. V příchozím požadavku je specifikován uživatel a příslušná podepsaná výzva pro autentizaci.
- **mobile_api#setDevice** – Slouží k registraci nového zařízení s unikátním klíčem zařízení *matchingKey*, veřejným klíčem *publickey*, identifikátorem Firebase *firebaseId* a přihlašovacím jménem *login*. Při platnosti umožňuje metoda registrovat nového uživatele pro vícefaktorovou autentizaci.

Sekvenční návrh operací na serveru

Na obrázku 3.8 je vyobrazen sekvenční diagram pro přihlášení na straně serveru. Tento diagram popisuje jednotlivé kroky a interakce, které se odehrávají při přihlašování uživatele. Průběh procesu je následující:

1. **Žádost o přihlášení:** Uživatel začne proces tím, že v přihlašovacím formuláři vyplní jméno a heslo, a potvrdí přihlášení.
2. **Spuštění vícefaktorové autentizace:** NextCloud zažádá ve třídě *WebMultifactorProvider* o funkci *getTemplate*, která spustí proces vícefaktorového ověření.
3. **Generování výzvy:** V dalším kroku zažádá *WebMultifactorProvider AplicationUserModel*, aby pro uživatele nastavil textovou výzvu *setSignTextChallenge*. Tuto výzvu následně vrací v návratové hodnotě.
4. **Získání firebaseID uživatele:** *WebMultifactorProvider* žádá *AplicationUserModel* o Firebase ID uživatele, které je potřebné pro odeslání notifikace. Tento údaj vrací v návratové hodnotě.
5. **Odeslání notifikace uživateli:** *WebMultifactorProvider* posílá notifikaci uživateli prostřednictvím funkce *SendNotification*, která předává Firebase ID, uživatelské jméno a vygenerovanou výzvu.
6. **Načtení šablony:** Na popředí aplikace se ve webovém prohlížeči zobrazí stránka s informací, že je třeba potvrdit přihlášení v chytrém zařízení. A je nastaven čas 60 vteřin pro tuto akci.

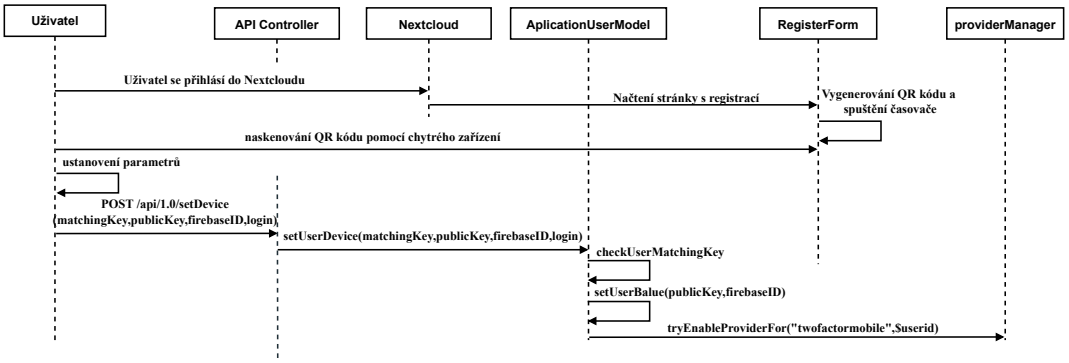


Obr. 3.8: Sekvenční model přihlášení na straně serveru.

7. **Uživatelské potvrzení:** Po potvrzení výzvy uživatelem nastane další fáze procesu, kdy je odeslán POST požadavek na *API Controller* se jménem uživatele a podepsanou výzvou.
8. **Ověření výzvy:** *API Controller* předá klíč do *AplicationUserModel*, který zjistí veřejný klíč uživatele a ověří podpis pomocí služby *SignatureVerifier*. Podle výsledku přihlášení se nastaví hodnota pro přihlášení na platnou.
9. **Zažádání o přihlášení do aplikace:** V dalším kroku se čeká na aplikaci až si zažádá o přihlášení což spustí funkci, která kontroluje hodnotu *setUserAllowLogin*, když je nastavená na platnou , tak uživatele vpustí dál do aplikace.
10. **Nastavení základních hodnot:** Po celém tomto procesu je nutné všechny vygenerované hodnoty nastavit zpátky na jejich počáteční hodnotu.

Proces registrace je na straně serveru výrazně jednodušší než samotné přihlášení. Návrh registrace je zobrazen na diagramu 3.9. Popis kroků je následující:

1. **Přihlášení do NextCloudu:** Uživatel se přihlásí do NextCloudu pomocí svých standardních přihlašovacích údajů.
2. **Načtení stránky s registrací:** Jakmile je uživatel úspěšně přihlášen, může přejít na stránku s registrací a začít průběh registrace kliknutím na tlačítko *registrovat*.



Obr. 3.9: Sekvenční model registrace na straně serveru.

3. **Generování QR kódu a spuštění časovače:** Na základě této interakce se vygeneruje QR kód. Tento kód je vyobrazen s parametry uvnitř něj na obrázku 3.10. Tyto parametry jsou název uživatele a unikátní vygenerovaný kód, který je potřeba pro ověření identity při odeslání POST požadavku v kroku číslo 6.

{"secretCode":"115376b4e0f066e28f854da5278d2
78e2d00e9279e09e2df6b1d7","user":"vedouci"}

Obr. 3.10: QR kód s daty.

4. **Naskenování QR kódu:** Uživatel naskenuje QR kód svým chytrým zařízením a na základě toho telefon vygeneruje potřebné údaje.
5. **Odeslání požadavku na registraci:** Je odeslán POST požadavek na *API Controller* NextCloudu s parametry matchingKey, publicKey, firebaseId, login.
6. **Kontrola identifikačního kódu:** Proběhne kontrola, jestli přijatý matchingKey souhlasí s vygenerovaným kódem z kroku 3.
7. **Ustanovení parametrů** Po úspěšné kontrole se nastaví v NextCloudu pro uživatele jeho veřejný klíč, firebaseId a dojde k registraci u *providerManager*, který aktivuje dvoufaktorové přihlášení.

4 Registrace a přihlášení z pohledu uživatele

Tato kapitola se věnuje prezentaci praktických výsledků práce. Výsledky budou prezentovány z pohledu uživatele. Hlavní pozornost je kladena na dva klíčové aspekty a to proces registrace služby a průběh přihlášení. Kromě těchto základních funkcionalit budou představeny další prvky, které byly v rámci práce vytvořeny, například registrace serveru.

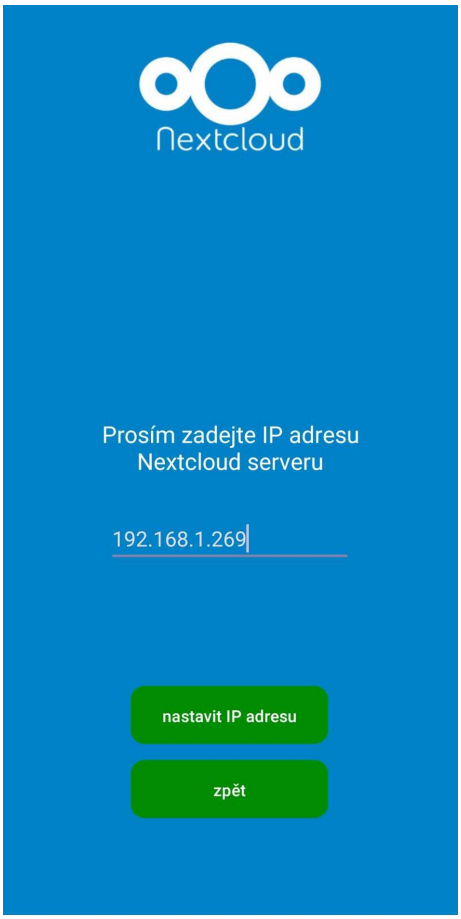
4.1 Registrace serveru

V této podkapitole je představena služba registrace serveru. Tato služba byla vytvořena jako dodatek k aplikaci, protože během vývoje aplikace pro vícefaktorovou autorizaci pro NextCloud byl server vždy hostován na lokálním zařízení. Mobilní aplikace proto musela zasílat POST požadavky na lokálně přidělenou adresu prostřednictvím DHCP (Dynamic Host Configuration Protocol) serveru. Změna adresy způsobovala nutnost manuální opravy v kódu, což představovalo značnou komplikaci. Pro zjednodušení správy řešení je tedy implementována funkce pro registraci serveru v grafickém rozhraní aplikace. Toto nastavení umožňuje uživateli dynamicky registrovat server podle aktuální IP adresy. Registrace probíhá následujícím způsobem:

V hlavním menu aplikace se na dolní straně nachází dvě tlačítka 4.1. Po stisknutí tlačítka *Nastavení serveru* dojde k přesměrování na obrazovku zobrazenou na obrázku 4.2.

Obr. 4.1: Hlavní obrazovka aplikace.

Na této stránce pak stačí vyplnit číslo IP adresy a potvrdit ji tlačítkem *nastavit IP adresu*. V rámci registrace je na pozadí vytvořena funkce pro kontrolu vstupního řetězce znaků, aby nedocházelo k zadání chybného formátu, či v případě překlepu byl uživatel informován. Po následném stisknutí tlačítka se zobrazí okno s oznámením, jestli nastavení proběhlo v pořádku, či došlo k nějaké chybě. Aplikace je nastavena pro port 8443, takže v případě hostování na standardním portu 443 je třeba změnit port v kódu.



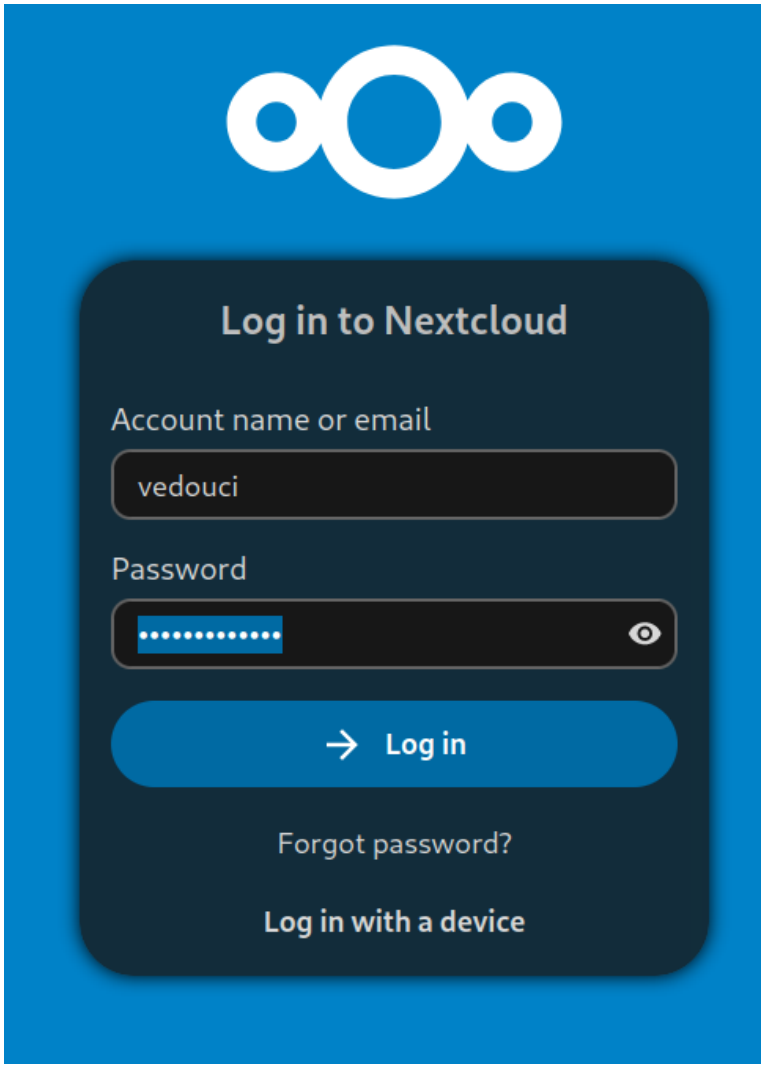
Obr. 4.2: Registrace IP adresy NextCloud serveru.

Navzdory výhodám, které dynamická registrace serveru přináší, je důležité vzít v úvahu i její nevýhody. Z bezpečnostního hlediska a pro zajištění stability provozu v produkčním prostředí je doporučeno preferovat statické nastavení konfigurace serverů přímo ve zdrojovém kódu. Dynamická registrace může být užitečná při testování nebo vývoji, ale v produkčním nasazení je vhodné dbát na pečlivou správu a kontrolu změn. Je to z toho důvodu, aby se minimalizovala rizika spojená s neautorizovanými nebo chybnými úpravami.

4.2 Registrace zařízení

V této podkapitole bude popsán proces registrace vícefaktorového ověření z pohledu uživatele. V samotném procesu registrace není zahrnuta instalace a nastavení oprávnění mobilní aplikace, jelikož se tento proces napříč různými zařízeními může lišit. Po nainstalování a nastavení aplikace je proces registrace následovný: Jako první se uživatel přihlásí do NextCloud webové aplikace standardními přihlašovacími údaji.

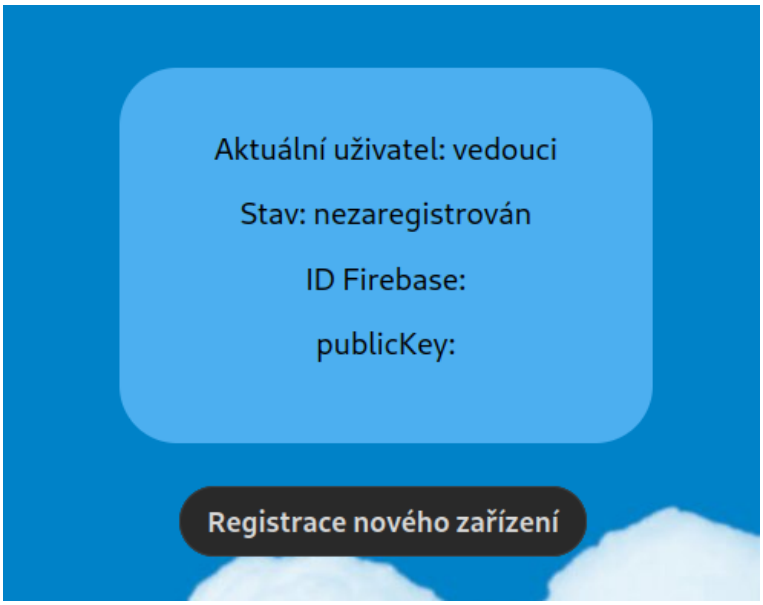
Formulář pro přihlášení je zobrazen na obrázku 4.3. Po úspěšném přihlášení je uži-



Obr. 4.3: Webová stránka pro přihlášení v NextCloudu.

vatel přesměrován na hlavní stránku aplikace. V levém horním rohu se nachází navigační menu, kde lze pomocí ikonky telefonu označené jako *Twofactor Mobile* pokračovat na registrační stránku vizualizovanou na obrázku 4.4. Na této stránce jsou zobrazeny základní údaje o uživateli, jeho přihlašovací jméno, stav registrace vícefaktorové autentizace, potenciální Firebase ID a veřejný klíč pro ověření výzev při přihlášení. Pro nový účet bez registrované služby bude informační tabule vypadat jako na obrázku 4.4. Tato tabule je vytvořena pro usnadnění vývoje a znázornění funkčnosti aplikace. V produkčním prostředí by neměla obsahovat osobní informace jako Firebase ID a veřejný klíč uživatele.

Dále je nutné v prohlížeči přejít tlačítkem *Registrace nového zařízení* na okno, které obsahuje podrobné informace o tom jak by měl daný uživatel postupovat.



Obr. 4.4: Informace o uživateli před registrací.

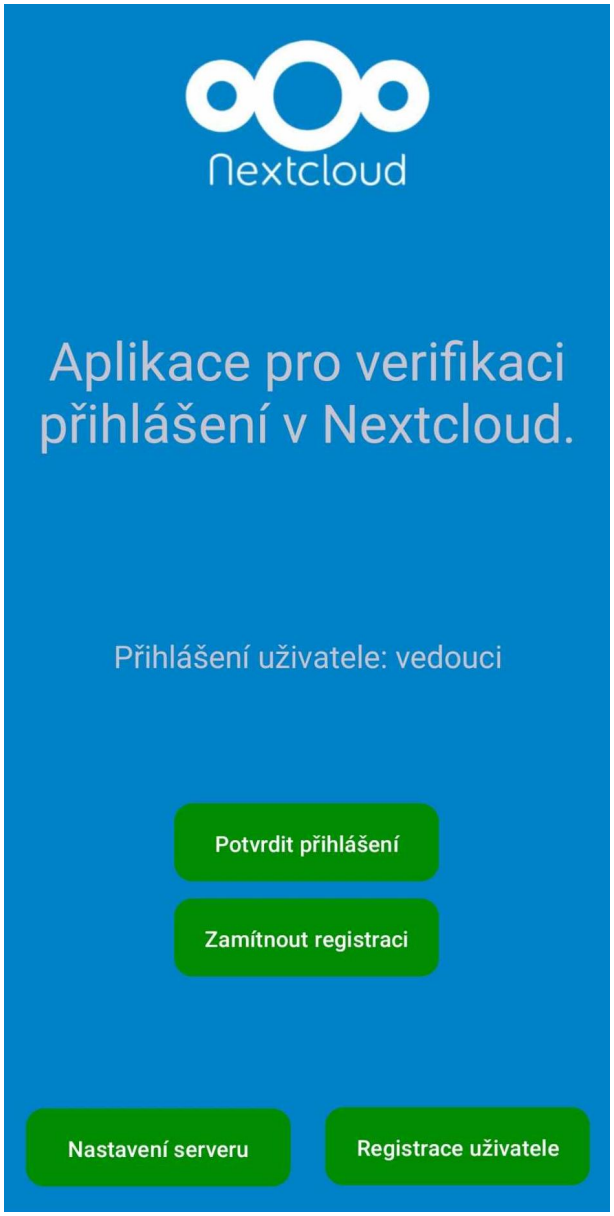
Obr. 4.5: Okno s QR kódem.

Další krok zahrnuje samotnou registraci. Uživatel si připraví mobilní telefon, ve kterém se při vstupu do aplikace autentizuje. Aplikace je přizpůsobena pro více-faktorovou autentizaci pomocí skenu otisku prstu. V aplikaci následně přejde na záložku registrace. Toho je dosaženo pomocí tlačítka *Registrace uživatele*, na hlavní stránce mobilní aplikace. Tato záložka obsahuje zobrazení fotoaparátu, kterým bude

třeba naskenovat příslušný kód. Tento kód se vygeneruje pomocí tlačítka *Generovat QR kód* na webové stránce NextCloud aplikace 4.5. Po úspěšném naskenování již budou v okně 4.4 dostupné přidělené údaje uživatele.

4.3 Přihlášení v aplikaci

Tato podkapitola popisuje proces přihlášení po aktivaci vícefaktorové autentizace.



Obr. 4.6: Potvrzení přihlášení v aplikaci.

Proces přihlášení probíhá následovně: Uživatel se standardně přihlásí do webové

aplikace pomocí svého uživatelského jména a hesla. V tuto chvíli může být použito snadno zapamatovatelné heslo, jako datum narození, protože potenciálnímu útočníkovi nestačí znát jen heslo. Musí mít také chytré zařízení, na kterém je účet registrován, a biometrické údaje pro autentizaci v tomto zařízení.

Po úspěšném přihlášení na webu je odeslána notifikace na mobilní zařízení uživatele. Tato notifikace upozorní na nutnost dokončení procesu přihlášení pomocí mobilní aplikace. Notifikace se může zobrazit na notifikační liště i když je aplikace vypnutá. Po otevření je vyzván k autentizaci pomocí otisku prstu. Tím je zajištěno ověření druhého faktoru a tím pádem k zajištění bezpečnosti v procesu přihlášení.

Po úspěšné autentizaci otiskem prstu se v aplikaci objeví tlačítko pro potvrzení přihlášení jako na obrázku 4.6. Uživatel si může vybrat jestli chce přihlášení pro daného uživatele potvrdit, nebo zamítnout. Po úspěšném odeslání odpovědi dojde na hlavní stránce k informování o stavu autentizace. Následně jsou tlačítka pro potvrzení a zamítnutí skryta. Nakonec je uživatel po zdařilé autentizaci během několika vteřin vpuštěn do NextCloud úložiště.

Závěr

Hlavním cílem práce bylo nastudovat problematiku současných metod vícefaktorového ověření u moderních ICT systémů. Na základě studie měl být následně navržen a implementován přístupový systém pro přístup do cloudové služby DECT. Systém měl být navržen tak, ať využívá Android mobilní aplikaci s konceptem vícefaktrvé autentizace na tomto zařízení.

První kapitola této práce obsahuje teorii potřebnou k porozumnění práce. Jsou v ní obsaženy informace o Autorizaci, Autentizaci, řízení přístupu a základům kryptografie. V rámci této části je taktéž probrán úvod do Multi-party a threshold signatures. Na tuto kapitolu navazuje průzkum řešení dostupných na nynějším trhu. Pozornost je kladena především na funkcionalitu, použité kryptografické protokoly, komunikační rozhraní, klíčový management a použité autentizační prvky. V rámci kapitoly je popsáno několik řešení a jáké přináší z hlediska procesu autentizace výhody a nevýhody.

Na tyto první teoretické kapitoly navazuje praktická realizace, která zahrnuje vývoj mobilní Android aplikace a implementaci serverové části aplikace pro DECT. V rámci praktické části bylo navrženo a realizováno řešení, které integruje vícefaktorovou autentizaci pomocí mobilní aplikace. Aplikace je založena na kryptografii veřejných klíčů a využívá pro podpis protokol ECDSA (Elliptic Curve Digital Signature Algorithm). V poslední kapitole je pak praktické vysvětlení principu registrace a přihlášení uživatele.

Výsledkem práce je funkční proces přihlášení, vytvořená mobilní aplikace a aplikace pro Nextcloud. Dosažené výsledky představují významný přínos pro zvýšení bezpečnosti při přístupu ke cloudové službě. Tento způsob ověření zcela eliminuje útoky hrubou silou a výrazně znesnadňuje phishingové útoky. Služba rovněž zvyšuje uživatelský komfort, protože eliminuje nutnost pamatovat si složitá hesla.

Tuto implementaci je možné dále rozšiřovat o další bezpečnostní prvky a metody autentizace, jako je sken obličeje nebo ověření pomocí hlasu. V budoucnu je také možné uvažovat o úplném odstranění tradičních přihlašovacích údajů, uživatelského jména a hesla, a přejít na autentizační metody založené výhradně na ověření druhého faktoru. Tento přístup by ještě více zjednodušil celý proces pro koncového uživatele.

Literatura

[1] TUMIN, Sharil a ENCHEVA, Sylvia. *A Closer Look at Authentication and Authorization Mechanisms for Web-based Applications.* [online]. Norsko: ResearchGate, 2012. ISBN 978-1-61804-089-3. [cit. 2023-09-29].
Dostupné z URL:
<https://www.researchgate.net/publication/250310860_A_Closer_Look_at_Authentication_and_Authorization_Mechanisms_for_Web-based_Applications>

[2] *Národní úřad pro kybernetickou a informační bezpečnost.* [online]. 2024. [cit. 2024-04-02].
Dostupné z URL:
<https://nukib.gov.cz/en/infoservis-en/publications-reports/>

[3] *New research: How effective is basic account hygiene at preventing hijacking.* [online]. 2019. [cit. 2024-04-02].
Dostupné z URL:
<https://security.googleblog.com/2019/05/new-research-how-effective-is-basic.html>

[4] TREVINO, Aranza. *Types of Multi-Factor Authentication (MFA).* [online]. 2023. [cit. 2023-11-18].
Dostupné z URL:
<https://www.keepersecurity.com/blog/2023/06/27/types-of-multi-factor-authentication-mfa/>

[5] TITTERINGTON, Alanna. *Types of two-factor authentication. Online.* [online]. [cit. 2024-04-02].
Dostupné z URL:
<https://www.kaspersky.com/blog/types-of-two-factor-authentication/48446/>

[6] MOROWCZYNSKI, Mark a EPPING, Michael. *Authentication and Authorization.* [online]. 2021, s. 10. [cit. 2023-11-28].
Dostupné z URL:
<https://www.researchgate.net/publication/359387651_Authentication_and_Authorization>

[7] PENELOVA, Maria. *Access Control Models. Online*. [online]. 2021, s. 28. [cit. 2023-11-28].
Dostupné z URL:
<https://www.researchgate.net/publication/356948823_Access_Control_Models>

[8] MAMMERI, Zoubir. *Cryptography Algorithms, Protocols, and Standards for Computer Security*. 1. Hoboken, New Jersey: John Wiley, 2023. ISBN 9781394207510.

[9] STALLINGS, William. Cryptography and Network Security: Principles and Practice. Online. Pearson Education Limited, 2017. ISBN 10:1-292-15858-1. [cit. 2024-05-14].
Dostupné z URL:
https://www.cs.vsb.cz/ochodkova/courses/kpb/cryptography-and-network

[10] NÁRODNÍ ÚŘAD PRO KYBERNETICKOU A INFORMAČNÍ BEZPEČNOST. Minimální požadavky na kryptografické algoritmy. Online. 2023. [cit. 2024-05-14].
Dostupné z URL:
<https://nukib.gov.cz/download/uredni_deska/Minimalni%20pozadavky%20na%20kryptograficke%20algoritmy.pdf>

[11] VALMIK, NehaKhatri a KSHIRSAGAR, V. K. Blowfish Algorithm. Online. 2014. ISSN 2278-8727. [cit. 2024-05-14].
Dostupné z URL:
<https://www.iosrjournals.org/iosr-jce/papers/Vol16-issue2/Version-10/N0162108083.pdf>

[12] JOURNAL OF INDUSTRIAL INFORMATION INTEGRATION. Asymmetric Cryptography. Online. 2022. [cit. 2024-05-14].
Dostupné z URL:
<https://www.sciencedirect.com/topics/computer-science/asymmetric-cryptography>

[13] TUCKER, Amanda. What is RSA Asymmetric Encryption? How Does it Work? Online. [cit. 2024-05-14].
Dostupné z URL:
<https://www.securew2.com/blog/what-is-rsa-asymmetric-encryption>

[14] What is ECDSA Encryption? How does it work? Online. [cit. 2024-05-14]. Dostupné z URL: <https://www.encryptionconsulting.com/education-center/what-is-ecdsa/>

[15] Elliptic Curve Cryptography. Online. 2023. [cit. 2024-05-14]. Dostupné z URL: <https://www.ibm.com/docs/en/ztpf/2022?topic=system-elliptic-curve-cryptography>

[16] IBM. Digital signature overview. Online. 2021. [cit. 2024-05-15]. Dostupné z URL: <https://www.ibm.com/docs/en/b2badv-communication/1.0.0?topic=overview-digital-signature>

[17] Co je to PKI? Online. 2021. [cit. 2024-05-15]. Dostupné z URL: <https://www.ssl.com/cs/co-je-pki>

[18] CLOUDFLARE, INC. What is SSL? Online. 2023. [cit. 2024-05-15]. Dostupné z URL: <https://www.cloudflare.com/learning/ssl/how-does-ssl-work>

[19] MULDER, Valentin; MERMOUD, Alain; LENDERS, Vincent a TELLENBACH, Bernhard. Trends in Data Protection and Encryption Technologies. Springer, 2023. ISBN 978-3-031-33385-9.

[20] COINBASE. What is a Multi-Party Computation (MPC) wallet? Online. 2024. [cit. 2024-05-15]. Dostupné z URL: https://www.coinbase.com/learn/wallet/what-is-a-multi-party

[21] Komentář ČBA k implementaci PSD2. Online. 2019. [cit. 2024-05-17]. Dostupné z URL: <https://cbaonline.cz/komentar-cba-k-implementaci-psd2>

[22] ČESKÁ SPOŘITELNA. Uživatelská příručka internetového a mobilního bankovnictví. Online. [cit. 2024-05-17]. Dostupné z URL: <https://www.csas.cz/static_internet/cs/Redakce/Ostatni/Ostatni_IE/Prilohy/up-george.pdf>

[23] ČESKÁ SPOŘITELNA. Uživatelská příručka aplikace George klíč. Online. [cit. 2024-05-17].
Dostupné z URL:
<https://www.csas.cz/static_internet/cs/Redakce/Ostatni/Ostatni_IE/Prilohy/up-george-klic.pdf>

[24] ČESKÁ SPOŘITELNA. Autentizace a autorizace uživatele. Online. [cit. 2024-05-17].
Dostupné z URL:
<https://developers.erstegroup.com/docs/tutorial/csas-how-to-call-api>

[25] GOOGLE LLC. Passwordless login with passkeys. Online. 2024. [cit. 2024-05-17].
Dostupné z URL:
<https://developers.google.com/identity/passkeys>

[26] UserVerification deep dive. Online. 2024. [cit. 2024-05-18].
Dostupné z URL:
<https://web.dev/articles/webauthn-user-verification>

[27] FACEBOOK. Co je bezpečnostní klíč a jak funguje? Online. [cit. 2024-05-18].
Dostupné z URL:
<https://www.facebook.com/help/401566786855239>

[28] SecurID Hardware Token. Online. 2022. [cit. 2024-05-20].
Dostupné z URL:
<https://help.access.securid.com/EN_US/Content/Production/ngx_c_hardware_token.html>

[29] NETSCALER. ECDSA cipher suites support. Online. 2023. [cit. 2024-05-20].
Dostupné z URL:
<https://docs.netscaler.com/en-us/citrix-adc/current-release/ssl/ciphers-available-on-the-citrix-ADC-appliances/ecdsa-cipher-suite-support-on-mpx-appliances.html>

[30] ANVIL. Using hardware tokens for two-factor authentication: how does it work? Online. 2024. [cit. 2024-05-20].
Dostupné z URL:
<https://anvil.works/blog/two-factor-auth-with-hardware>

[31] ADOBE SYSTEMS. User Authentication. Online. 2023. [cit. 2024-05-20]. Dostupné z URL: <https://developer.adobe.com/developer-console/docs/guides/authentication/UserAuthentication/>

[32] *TELEPORT. How OAuth 2.0 Works*. Online. 2022. [cit. 2023-12-13]. Dostupné z URL: <https://goteleport.com/blog/how-oauth-authentication-works/>

[33] FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT. OAuth 2.0. Online. 2023. [cit. 2024-05-20]. Dostupné z URL: <https://help.fit.cvut.cz/dev/oauth2.html>

[34] The OAuth 2.0 Authorization Framework. Online. 2012. [cit. 2024-05-20]. Dostupné z URL: <https://www.rfc-editor.org/rfc/pdfrfc/rfc6749.txt.pdf>

[35] DIGITAL INFORMATION WORLD. Should You Use OAuth 2.0? Pros and Cons. Online. 2023. [cit. 2024-05-20]. Dostupné z URL: <https://www.digitalinformationworld.com/2023/11/should-you-use-oauth-20-pros-and-cons.html>

[36] Introducing Public Key Cryptography and Web Authentication (WebAuthn). Online. [cit. 2024-05-08]. Dostupné z URL: <https://webauthn.guide/#about-webauthn>

[37] *WebAuthn 101: How Web Authentication Works*. Online. 2022. [cit. 2023-12-13]. Dostupné z URL: <https://www.descope.com/learn/post/webauthn>

[38] *NEXTCLOUD. Nextcloud latest user manual introduction*. Online. [cit. 2023-12-13]. Dostupné z URL: <https://docs.nextcloud.com/server/latest/user_manual/en/>

[39] *Řízení přístupu ke cloudové službě pomocí čipových karet*. Online. [cit. 2024-04-29]. Diplomová práce. Brno: Vysoké učení technické v Brně, 2022. Dostupné z URL: <https://www.vut.cz/studenti/zav-prace/detail/141407>

[40] *NEXTCLOUD*. App store. Online. 2016. [cit. 2024-04-29].
Dostupné z URL:
<https://apps.nextcloud.com/>

[41] GOOGLE LLC. Firebase. Online. [cit. 2024-05-01].
Dostupné z URL:
<https://firebase.google.com/docs/cloud-messaging>

[42] Pusher vs. Firebase. Online. 2023. [cit. 2024-05-02].
Dostupné z URL:
<https://ably.com/topic/pusher-vs-firebase>

[43] Manifest.permission. Online. [cit. 2024-05-14].
Dostupné z URL:
<https://developer.android.com/reference/android/Manifest.permission>

[44] NAZIRIDIS, Nick. Porovnání ECDSA vs RSA. Online. 2018. [cit. 2024-05-06].
Dostupné z URL:
<https://www.ssl.com/cs/%C4%8Dl%C3%A1nek/porovn%C3%A1n%C3%AD-ecdsa-vs-rsa/>

[45] Android Keystore system. Online. 2024. [cit. 2024-05-07].
Dostupné z URL:
<https://developer.android.com/privacy-and-security/keystore>

[46] BISCEGLIA, Noah. *Technology: How Do Passkeys Work?*. [online]. 2023. [cit. 2023-11-21].
Dostupné z URL:
<https://teampassword.com/blog/passkey-technology>

Seznam symbolů a zkratek

AES	Advanced Encryption Standard
API	application programming interface
DAC	Discretionary Access Control
DDOS	Distributed denial of service
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DOS	Denial of service
DSA	Digital Signature Algorithm
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
HTTPS	Hypertext Transfer Protocol Secure
ID	Identification
JSON	JavaScript Object Notation
MAC	Mandatory Access Control
MPC	Multi-party Computation
NIST	National Institute of Standards and Technology
NUKIB	Národní úřad pro kybernetickou a informační bezpečnost
PIN	Personal identification number
PKI	Public key infrastructure
PSD2	Payment Services Directive 2
QR	Quick Response
RBAC	Role Based Access Control
RSA	Rivest, Shamir, Adleman
SCA	Strong Customer Authentication

SDK	Software Development Kit
SMS	Short message service
SSL	Secure Sockets Layer
TC	Threshold Cryptography
TLS	Transport Layer Security
TOTP	Time-based One-time Password
USB	Universal serial bus
VoIP	Voice over Internet Protocol

A Seznam kódu

Výpis A.1: Odeslání dat na Firebase server.

```
$mobData = json_decode(file_get_contents
    ('/var/www/html/custom_apps/twofactormobile/mob.json'), true);
```

Výpis A.2: Data, která obsahuje výzva při odeslání.

```
"data":{
    "title": "Prosím autorizujte se v aplikaci",
    "body": "' . $userName . '",
    "challenge": "' . $challenge . '"
},
```

Výpis A.3: Oprávnění v aplikaci.

```
uses-permission android:name="android.permission.INTERNET
uses-permission android:name="android.permission.
    ↪ ACCESS_NETWORK_STATE
uses-permission android:name="android.permission.CAMERA
uses-permission android:name="android.permission.USE_BIOMETRIC
uses-permission android:name="android.permission.USE_FINGERPRINT
uses-permission android:name="android.permission.FOREGROUND_SERVICE
```

Výpis A.4: Externí knihovny použité v projektu.

```
implementation(platform("com.google.firebase:firebase-bom:32.7.4"))
implementation("com.google.firebase:firebase-messaging")
implementation("com.google.firebase:firebase-analytics")
implementation("com.squareup.okhttp3:okhttp:4.9.3")
implementation("com.google.zxing:core:3.4.1")
implementation("com.journeyapps:zxing-android-embedded:4.2.0")
implementation("com.google.android:play-services-vision:20.1.3")
implementation("org.bouncycastle:bcpkix-jdk15on:1.69")
implementation("androidx.biometric:biometric:1.1.0")
```

Výpis A.5: Generování klíčů pomocí ECDSA.

```
public void generateKeyPair(String alias) {
    try {
        KeyPairGenerator kpg = KeyPairGenerator.getInstance(
            KeyProperties.KEY_ALGORITHM_EC, "AndroidKeyStore");
        kpg.initialize(new KeyGenParameterSpec.Builder(
            alias,
            KeyProperties.PURPOSE_SIGN | KeyProperties.
                ↪ PURPOSE_VERIFY)
            .setDigests(KeyProperties.DIGEST_SHA256,
                    KeyProperties.DIGEST_SHA512)
            .build());

        KeyPair kp = kpg.generateKeyPair();
    } catch (NoSuchAlgorithmException | NoSuchProviderException |
            InvalidAlgorithmParameterException e) {
        e.printStackTrace();
    }
}
```

B Obsah elektronické přílohy

- `/` ..... kořenový adresář přiloženého archivu
  - `github_linky.txt` ..... odkaz na github repozitáře projektu
  - `MobileTwofactorDect` ..... Kód mobilní aplikace
    - `app`
      - `drawable` ..... Grafický návrh stránky
        - `custom_button_background.xml`
        - `ic_launcher_background.xml`
        - `ic_launcher_foreground.xml`
        - `ic_notification.png`
        - `nextcloud_logo.jpg`
      - `mobiletwofactordect` ..... Implementované třídy v Javě
        - `ECDSAKeyManager.java`
        - `GetTokenForApp.java`
        - `MainActivity.java`
        - `MyFirebaseMessagingService.java`
        - `RegisterPhone.java`
        - `SendRegistrationParams.java`
        - `SetupServer.java`
    - `readme.md`
  - `twofactormobile` ..... Kód Nextcloud aplikace
    - `js` ..... Javascriptové soubory
      - `login.js`
      - `registerFormLogic.js`
    - `Controller` ..... Kontrolery aplikace
      - `MobileApiController.php`
      - `NoteApiController.php`
      - `NoteController.php`
      - `PageController.php`
      - `WebSocket.php`
    - `Provider` ..... Poskytovatelé služeb
      - `WebMultifactorProvider.php`
    - `Service` ..... Služby v aplikaci, odeslání notifikace, práce s podpisy a další
      - `AplicationUserModel.php`
      - `SendNotification.php`
      - `SignatureVerifier.php`
      - `WebSocketServer.php`
    - `templates` ..... Šablony pro frontend aplikace
      - `PhoneMultifactorChallenge.php`
      - `RegisterForm.php`