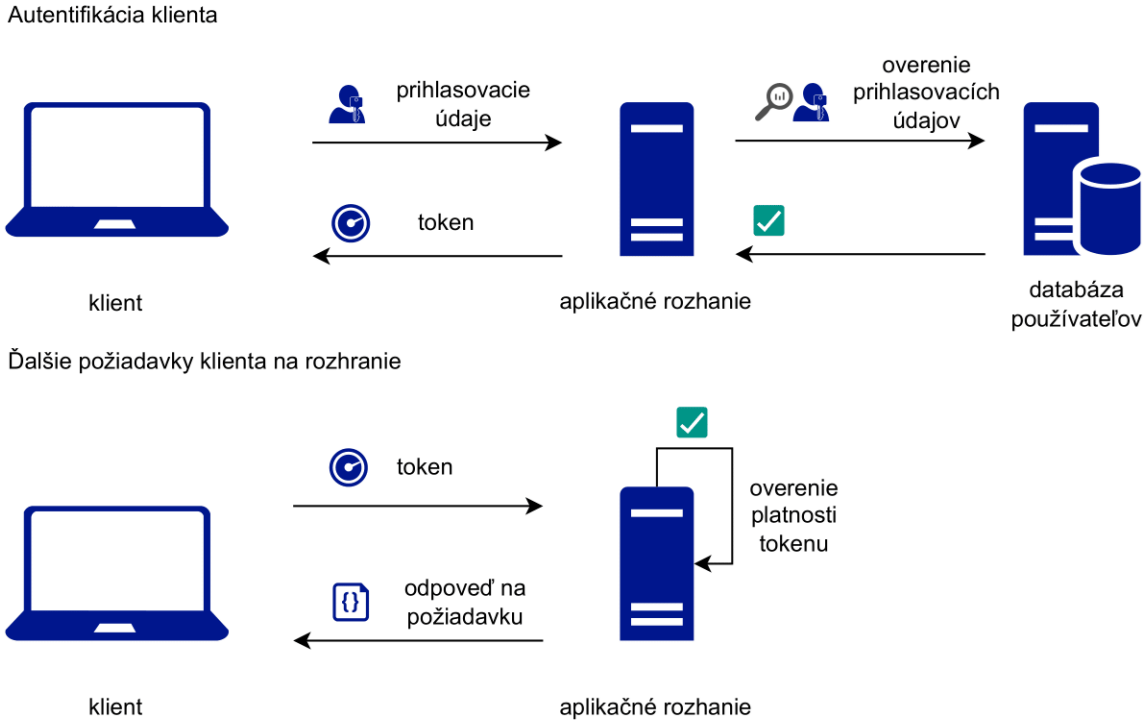
Porovnanie API tokenov

Jitka Muravská, školiteľ: RNDr. Richard Ostertág, PhD.

API token v schéme zabezpečenia



Dôležité aspekty API tokenu

- Prenášané dáta
  - identifikátor používateľa
  - prístupové práva
  - platnosť tokenu
- Bezpečnosť
  - ochrana autentickosti, integrity
  - rôzne zraniteľnosti
- Rýchlosť

Priebeh a výsledky práce

- Študovanie špecifikácií tokenov
  - formát, generovanie, validácia, zapojenie tretích strán do autorizácie
- Teoretické porovnanie
  - bezpečnosť, flexibilita a popularita
- Praktické porovnanie na testovacom API
  - rýchlosť

Porovnávané tokeny

- Nepriehľadný token - náhodný reťazec
- Štruktúrované tokeny
  - JWT
  - PASETO
  - Fernet
  - Branca
  - Macaroons
  - Biscuits

Teoretické porovnanie

Vlastnosť	Nepriehľadný	JWT	PASETO	Fernet	Branca	Macaroons	Biscuits
Počet kryptografických funkcií	1	30	6	1	1	1	1
Server neurčuje podpisový algoritmus iba z tokenu	-	○	○	●	●	●	●
Odolnosť voči útoku pomýlením algortimu	-	○	◐	●	●	●	●
Riešenie problému odvolania	●	○	○	○	○	◐	◐
Odolnosť voči útoku opakovaním	-	○	○	○	○	◐	◐
Ochrana dôvernosti	-	◐	◐	●	●	○	○
Overenie autentickosti a integrity hocikým	○	◐	●	●	●	●	●
Zoslabenie tokenu hocikým	-	-	-	-	-	●	●
Autorizačná schéma v tokene	-	◐	◐	◐	◐	●	●
Bezstavová validácia	-	●	●	●	●	●	●
Štandardná validácia	-	●	●	○	○	◐	●
Popularita	-	●	◐	◐	○	◐	○

Vybrané výsledky porovnania - bezpečnosť

Vlastnosť	Nepriehľadný	JWT	PASETO	Fernet	Branca	Macaroons	Biscuits
Riešenie problému odvolania	●	○	○	○	○	◐	◐
Odolnosť voči útoku opakovaním	-	○	○	○	○	◐	◐

- Problém odvolania a útok opakovaním
  - zneplatnenie vydaného tokenu serverom
  - nepriehľadný token - využije stav na serveri
  - štruktúrovaný token - bezstavový server
  - riešenie:
    - povoľovací a zamietací zoznam
      - Macaroons a Biscuits - zoznamy v stave služby tretej strany
    - krátka časová platnosť tokenov

Vybrané výsledky porovnania - bezpečnosť

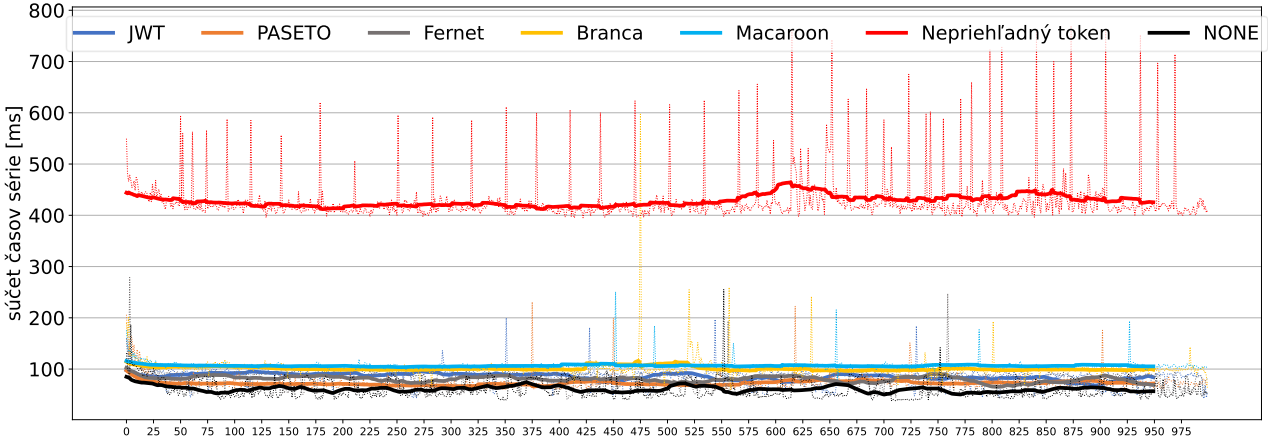
Vlastnosť	Nepriehľadný	JWT	PASETO	Fernet	Branca	Macaroons	Biscuits
Počet kryptografických funkcií	1	30	6	1	1	1	1
Server neurčuje podpisový algoritmus iba z tokenu	-	○	○	●	●	●	●
Odolnosť voči útoku pomýlením algortimu	-	○	◐	●	●	●	●

- Útok pomýlením algoritmu
  - Server validuje podpis tokenu iným algoritmom ako má
  - Nepriehľadný token, Fernet, Branca, Macaroons, Biscuits
    - používajú najviac 1 kryptografické primitívum
    - absolútne bezpečné
  - riešenie:
    - identifikátor kľúča v tokene
    - kontrola formátu kľúča

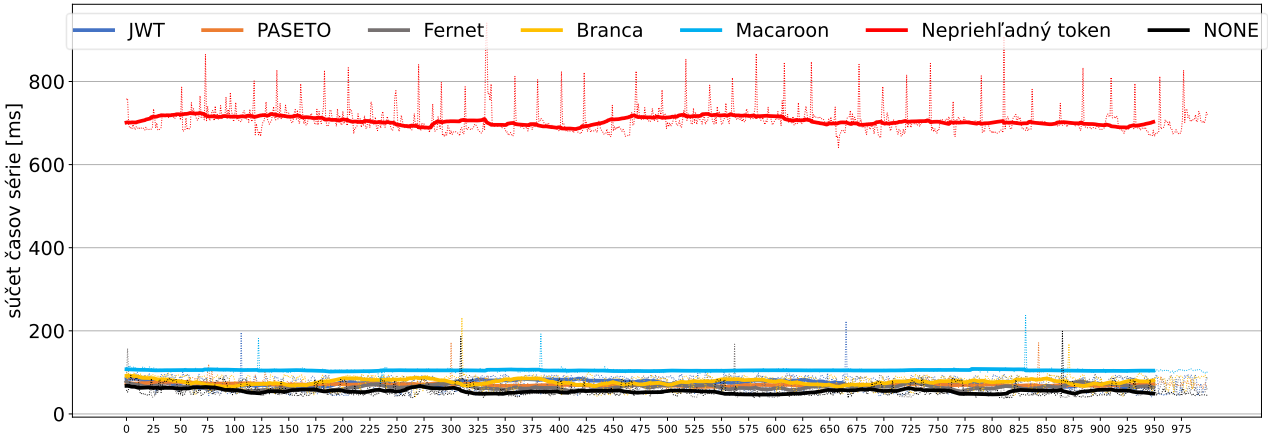
Praktické porovnanie na testovacom API

- Implementácia jednoduchého API
  - JavaScript, sqlite
  - bez Biscuits
- 2 koncové body
  - generovanie tokenu
  - validácia tokenu
- Implementácia klienta
- Meranie
  - 1000 sérií po 100 meraní
  - NONE token pre menerie réžie
  - Automatizované generovanie grafov pomocou Pythonu

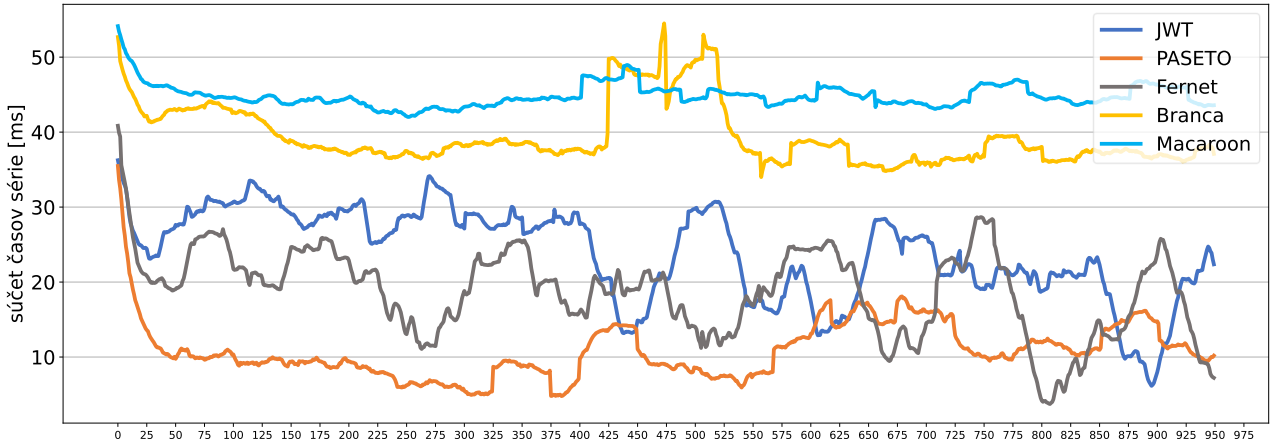
Generovanie tokenu



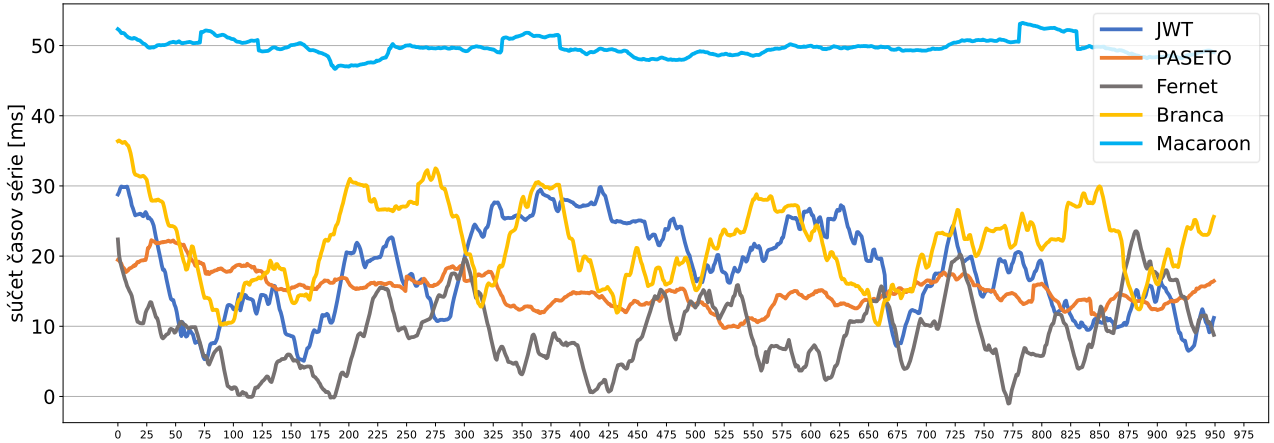
Validácia tokenu



Generovanie tokenu



Validácia tokenu



Zhrnutie

- Naštudovanie špecifikácií tokenov
- Navrhnutie metrík porovnania
- Porovnanie
- Implementácia testovacej aplikácie

Ako v budúcnosti nadviazať na prácu

- Porovnať ďalšie tokeny
  - napríklad tokeny, ktoré vzniknú v budúcnosti
- Analyzovať viaceré knižnice
- Rozšíriť testovacie API o ďalšie funkcie
  - napríklad pridať tretie strany do procesu autorizácie

„Návrh a realizácia experimentov v kapitole 4 pôsobia nepremyslene. Aký význam má zahŕňať do experimentu klienta a jeho komunikáciu (vrátane komunikácie servera)? Z hľadiska náročnosti konštrukcie a overovania tokenu je to irelevantné, resp. pre všetky tokeny rovnaké. Okrem toho tieto aktivity zvyšujú šum v nameraných dátach. V kapitole som márne hľadal rozdelenie nameraných časov na jednotlivé časti komunikácie. Ak bolo cieľom merať operácie aj s týmito časťami, nedáva to veľký zmysel, lebo v praxi bude málokedy prebiehať komunikácia lokálne. Nakoniec sa podľa nameraných údajov aj tak zdá, že operácie s testovanými API tokenmi sú dostatočne rýchle a toto nebude tým kritériom, ktoré by rozhodovalo pri výbere konkrétneho API tokenu.“

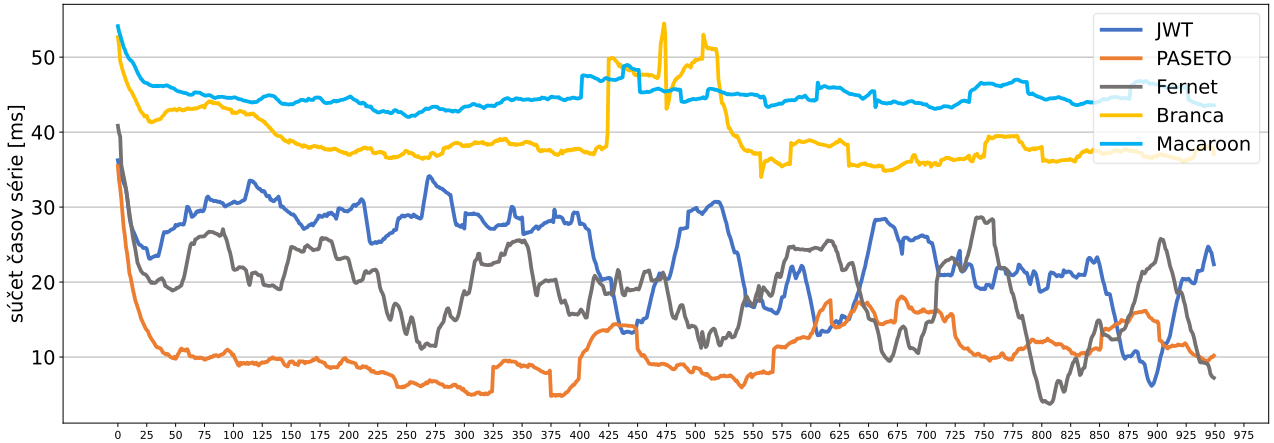
Testovanie pomocou klienta

- Návrh bol inšpirovaný štandardným testovaním webových aplikácií
  - využíva klienta
  - klient posiela maximálny počet požiadaviek na server
  - meria sa celý čas od vyslania požiadavky až po prijatie odpovede
- Implementované meranie s paralelnými klientami
  - ukázalo sa ako irelevantné

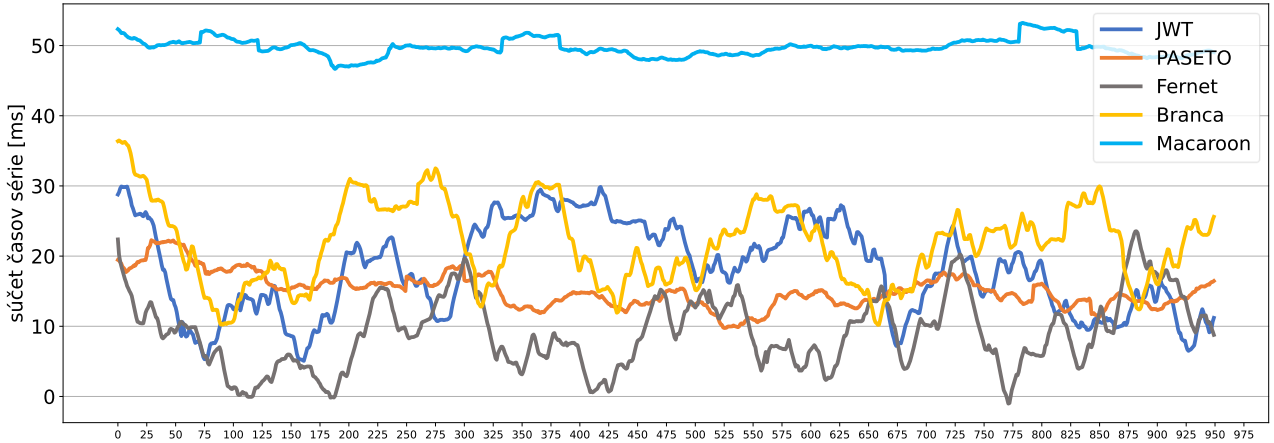
Rozdelenie nameraných časov

- Implementovaný NONE token
- Porovnanie rýchlosti s hodnotami po odpočítaní NONE tokenu

Generovanie tokenu



Validácia tokenu



„Toto tvrdenie v časti 2.2.1 na strane 15 je nezrozumiteľné, ako obyčajným vložením verejného kľúča pred podpísaním tokenu dosiahnuť nepopierateľnosť autorstva: Napríklad verzia v3 prináša nepopierateľnosť autorstva ako novú bezpečnostnú kvalitu. Dosahuje to dokonca bez predĺženia podpisu a to pomocou pridania verejného kľúča do tokenu pred vypočítaním podpisu [49]. “

- Podrobnejšie popísané v citovanom článku [49]
- Daná podpisová schéma (G, Σ, V) spĺňajúca isté predpoklady
- Vytvoríme schému (G’, Σ’, V’) poskytujúcu nepopierateľnosť autorstva:
  - $G' = G;$
  - $\Sigma'(K_{\text{priv}}, m) = \Sigma(K_{\text{priv}}, h(m||K_{\text{pub}}))$ (where $||$ denotes concatenation);
  - $V'(K_{\text{pub}}, m, s) = V(K_{\text{pub}}, h(m||K_{\text{pub}}), s).$

[49] Thomas Pornin and Julien P. Stern. Digital signatures do not guarantee exclusive ownership. http://www.bolet.org/~pornin/2005-acns-pornin+stern. pdf.

“Niekde by sa hodilo spresnenie textu, napr. v časti 1.6.1 na strane 8, kde v konštatovaní limit platnosti, ktorý býva kvôli bezpečnosti krátky chýba bližšia informácia o obvyklej hodnote, keďže predpokladám, že pre rôzne typy tokenov znamená krátky iný interval platnosti.”

- Časť 1.6.1 popisuje prístupový token

1.6.2 Nositeľský token

Nositeľský token reprezentuje najčastejší spôsob použitia prístupového tokenu. Pri jeho použití má nositeľ tokenu prístup k rozhraniu bez ohľadu nato, kto je. Stačí aby v požiadavke poslal platný nositeľský token.

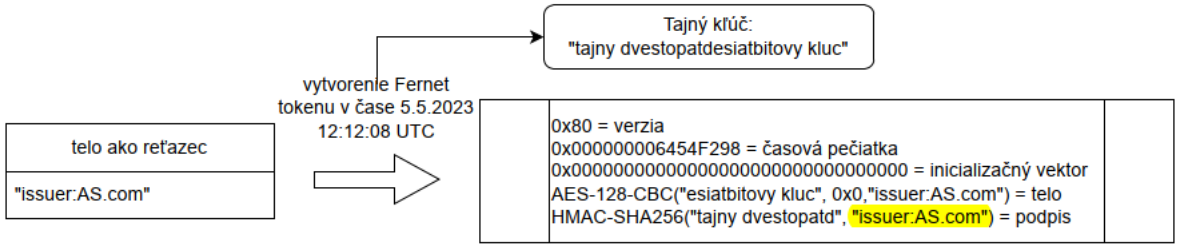
Následne rozhranie nekontroluje identitu používateľa, ale iba platnosť tokenu. Preto pri jeho využití treba prikladať väčší dôraz na bezpečnosť tokenu napríklad tak, že jeho platnosť bude veľmi krátka, napríklad 5 minút.

„(str. 32) V bakalárskej práci očakávam presnejšie vyjadrovanie ako napr.toto: V prípade podpisov tokeny používajú digitálne podpisy na základe asymetrického šifrovania...“

3.1.2 Útok pomýlením algoritmu

V prípade podpisov tokeny používajú digitálne podpisy na základe asymetrického šifrovania a teda dvojicu súkromného a verejného kľúča alebo hešovanie s kľúčom, ktorý je tajný. Jediný kľúč, ku ktorému má útočník ľahký prístup je verejný kľuč (označme ho pk). Útok pomýlením algoritmu potom prebehne tak, že útočník podpíše token hešova-

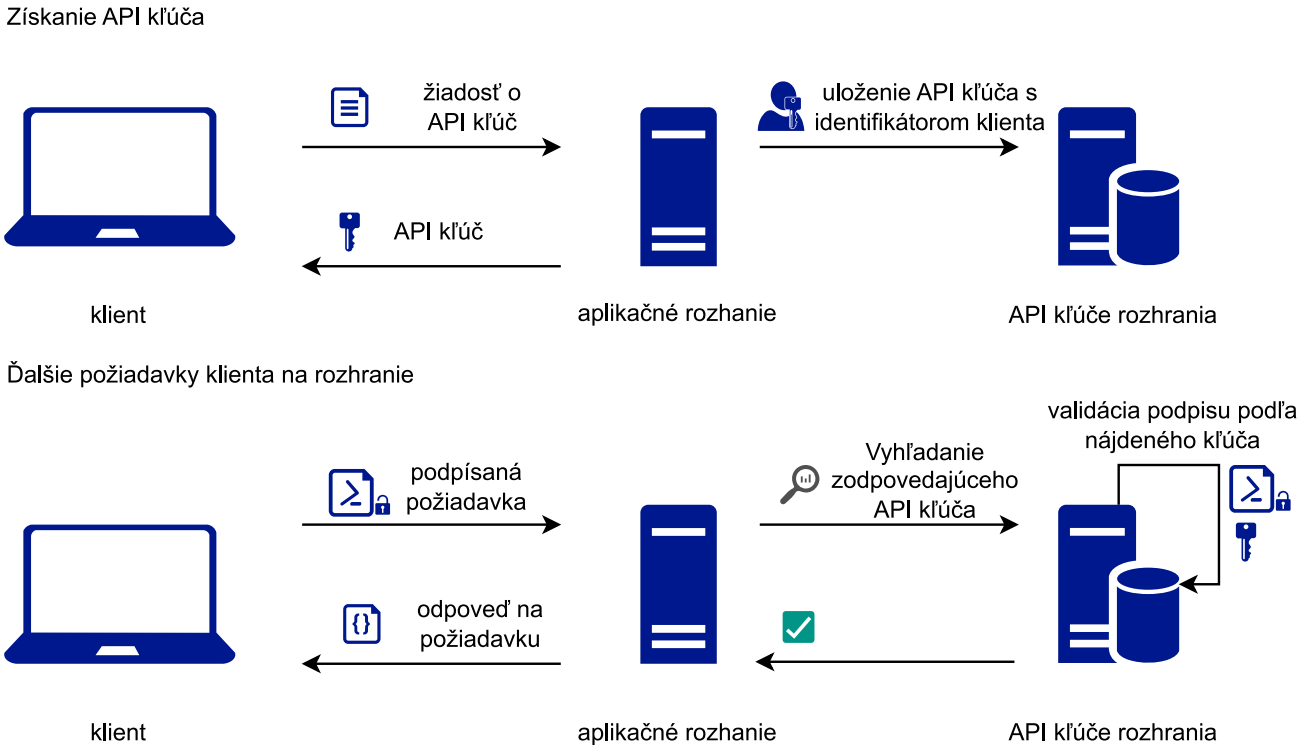
„Obrázok 2.3 na strane 18 je chybný, HMAC sa počíta zo zreťazenia verzie, časovej pečiatky, IV a šifrového textu, nie len z tela samotného. Podobne je chybný aj obrázok 2.4 na strane 19.“



Obr. 2.3: Príklad jednoduchého Fernet tokenu.

- Na obrázku je chyba
  - vznikla nedôslednosťou pri vyrábaní obrázku
- V texte práce je ale popísaný korektný výpočet HMAC

„Na obrázku 1.2 na strane 6 má podpísaná požiadavka pravdepodobne “tiecť” až na koniec, teda nielen vyhľadanie zodpovedajúceho API kľúča ale skôr ide o overenie podpísanej požiadavky.“



„Testovacia aplikácia je vlastne len meraním času pre operácie na rôznych API tokenoch. Práca, elektronická príloha ani verzia na github neobsahuje akékoľvek inštalačné inštrukcie a predpoklady pre spustenie implementovaných testov (a samozrejme, implementácia je platformovo špecifická).“

Meranie a interpretácia výsledkov

Meranie spustí server rozhrania postupne s každým tokenom a pre každý spustí klienta, ktorý vykoná 100000 požiadaviek na koncový bod `signin` a 100000 požiadaviek na koncový bod `welcome`. Na spracovanie prvej požiadavky musí rozhranie vygenerovať vybraný token, na spracovanie druhej musí validovať zaslaný token.

Meranie - namerané hodnoty zapíše do súboru `measure.out`:

```
npm run measure-all-single
```

Vygenerovanie grafov do priečinku `graphs` a HTML reportu do súboru `report.html` podľa nameraných hodnôt v súbore `measure.out`:

```
npm run graphs
```

Meranie a následné vygenerovanie grafov z nameraných hodnôt:

```
npm run run-graphs
```

Inštalácia Node.js a npm:

```
sudo apt install nodejs
sudo apt install npm
```

Inštalácia JS knižníc:

```
npm install
```

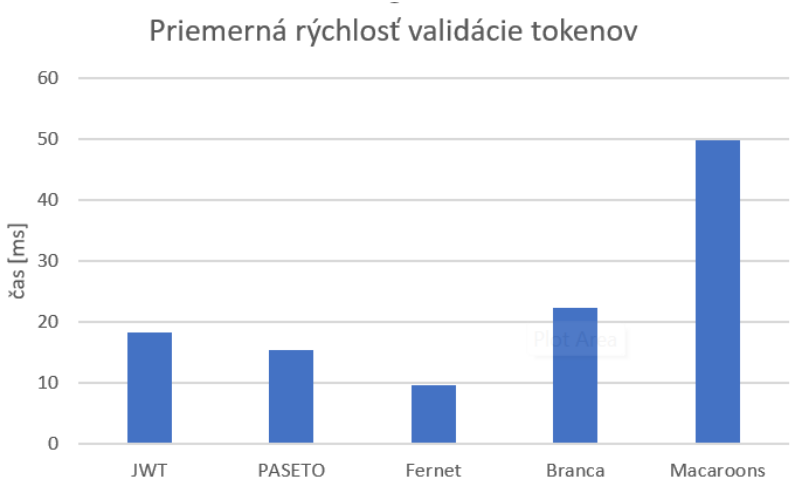
Inštalácia Python knižníc:

```
pip install matplotlib
pip install numpy
pip install Jinja2
```

- Python a Node.js sú platformovo závislé
- Samotný program je platformovo nezávislý
- Knižnice sú nainštalovateľné pomocou štandardných manažérov balíčkov (pip, npm)

„Grafy na obr. 4.11 a 4.12 sú úplne zbytočné, zobrazujú konštanty a lepšie by bolo tieto hodnoty zobraziť v tabuľke.“

- Grafická reprezentácia dáva jednoduchší prehľad
- Namiesto čiarového grafu by bol vhodnejší stĺpcový



„(kapitola 3) V grafoch hodnotiacich jednotlivé API tokeny by bolo lepšie, ak by rovnaké symboly vždy znamenali pozitívne resp. negatívne hodnotenie. V opačnom prípade sú tabuľky horšie interpretovateľné napr. náchylnosť na útoky vs. ochrana dôvernosti.“

Upravená tabuľka

Vlastnosť	Nepriehľadný	JWT	PASETO	Fernet	Branca	Macaroons	Biscuits
Počet kryptografických funkcií	1	30	6	1	1	1	1
Server neurčuje podpisový algoritmus iba z tokenu	-	○	○	●	●	●	●
Odolnosť voči útoku pomýlením algortimu	-	○	◐	●	●	●	●
Riešenie problému odvolania	●	○	○	○	○	◐	◐
Odolnosť voči útoku opakovaním	-	○	○	○	○	◐	◐
Ochrana dôvernosti	-	◐	◐	●	●	○	○
Overenie autentickosti a integrity hocikým	○	◐	●	●	●	●	●
Zoslabenie tokenu hocikým	-	-	-	-	-	●	●
Autorizačná schéma v tokene	-	◐	◐	◐	◐	●	●
Bezstavová validácia	-	●	●	●	●	●	●
Štandardná validácia	-	●	●	○	○	◐	●
Popularita	-	●	◐	◐	○	◐	○

Pôvodná tabuľka z práce

Vlastnosť	Nepriehľadný	JWT	PASETO	Fernet	Branca	Macaroons	Biscuits
Počet kryptografických funkcií	1	30	6	1	1	1	1
Určenie podpisového algoritmu z tokenu	⊘	●	●	○	○	○	○
Náchylnosť na útok pomýlením algortimu	⊘	●	◐	○	○	○	○
Riešenie problému odvolania	●	○	○	○	○	◐	◐
Náchylnosť na útok opakovaním	⊘	●	●	●	●	◐	◐
Ochrana dôvernosti	⊘	◐	◐	●	●	○	○
Overenie autentickosti a integrity hocikým	○	◐	●	●	●	●	●
Zoslabenie tokenu hocikým	⊘	⊘	⊘	⊘	⊘	●	●
Autorizačná schéma v tokene	⊘	◐	◐	◐	◐	●	●
Bezstavová validácia	⊘	●	●	●	●	●	●
Štandardná validácia	⊘	●	●	○	○	◐	●
Popularita	⊘	●	◐	◐	○	◐	○