Fachhochschule Kiel
Hochschule für Angewandte Wissenschaften

Passkeys – a sophisticated alternative to passwords

Prof. Dr.-Ing. Ulrich Jetzek

Kiel University of Applied Sciences, Germany

Institute of Communications Technology and Embedded Systems

International Symposium on

Ambient Intelligence and Embedded Systems

September 24th – 27th, 2025

Hamburg University of Applied Sciences

Hamburg, Germany

Overview

1. Introduction
2. Passwords – how do they work?
3. Disadvantages of passwords
4. FIDO – Fast IDentity Online – a short overview
5. Passkeys – how do the work?
6. A simple passkey example
7. Passkey Benefits
8. Conclusion

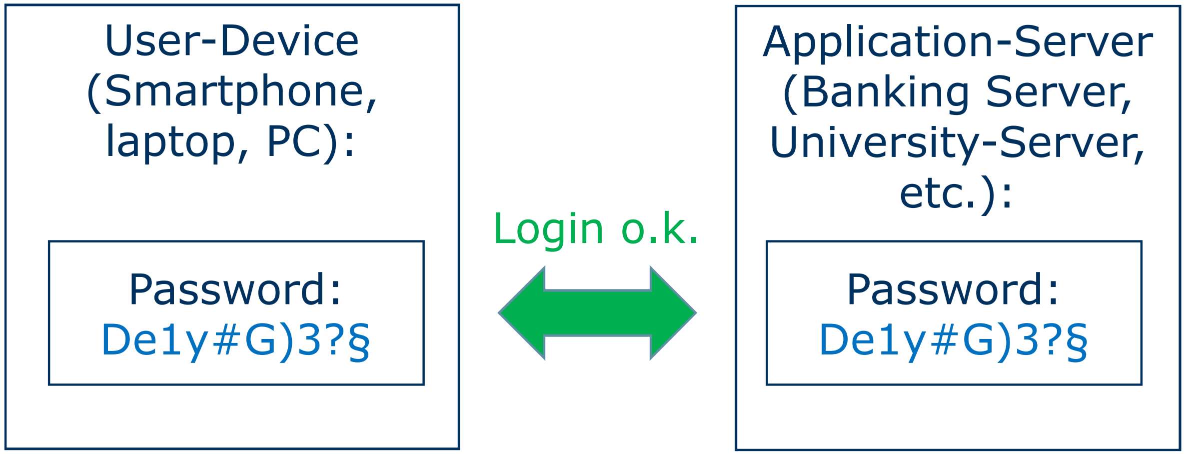
1. Introduction

- Passwords are common in many applications, but
  - Phishing or other attacks may lead to ‚stolen' passwords
  - Accounts can be attacked quite easily
- Problem for the individual as well as for institutions, companies, organizations …

- Do alternatives for passwords exist?
  - Multi-factor-authentication
  - Passkeys

- This talk shall give insight into the functionality and advantages of passkeys …
- Be curious …

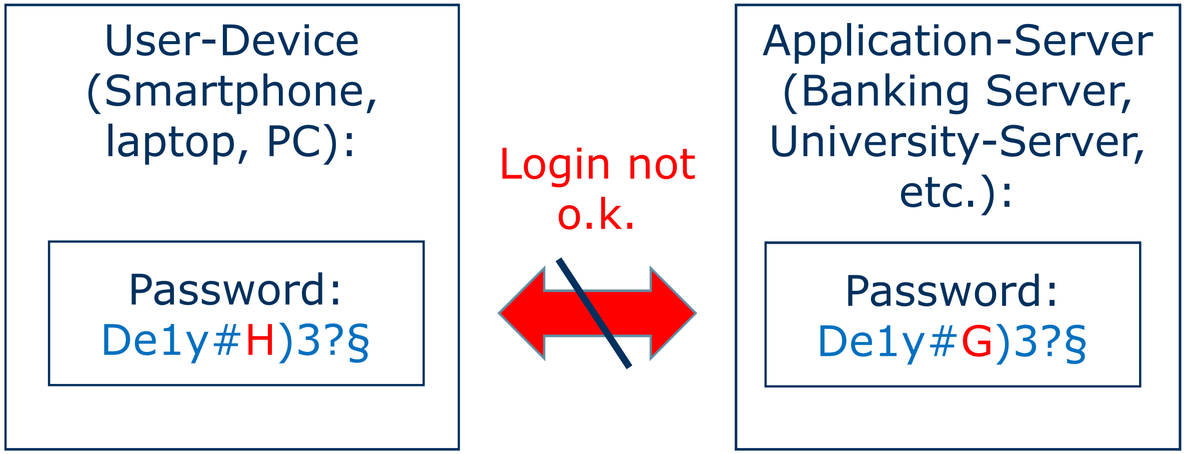
2. Passwords – How do they work



- Out of crypto-perspective:
- Usage of passwords corresponds to:

  symmetric crypto procedure
  - Sender and receiver use the **same** secret (private) key for encryption and decryption
  - Storage and transfer of passwords is – of course – done encrypted

3. Passwords – Major Problems



- Password entered by user is compared with the one stored on the application server

- If either
  - Password ‚phished' by third party OR
  - Password leak on application server

  password may be used by others (‚Oscar') without being noticed by application server

Fachhochschule Kiel

3. Password Usage and Disadvantages

- # of accounts / registrations for the individual continuously increasing …
  - University
  - Banking / assurances
  - Online shopping
  - Social media
  - …

- Registration in most cases: simple and a one-minute-process, but …
  - In many cases simple passwords are used, like: summer25, 12345, …
  - Often same password used for *various* accounts OR:
  - Many accounts = many passwords = difficult to remember all of them
  - In many cases passwords are kept for long time

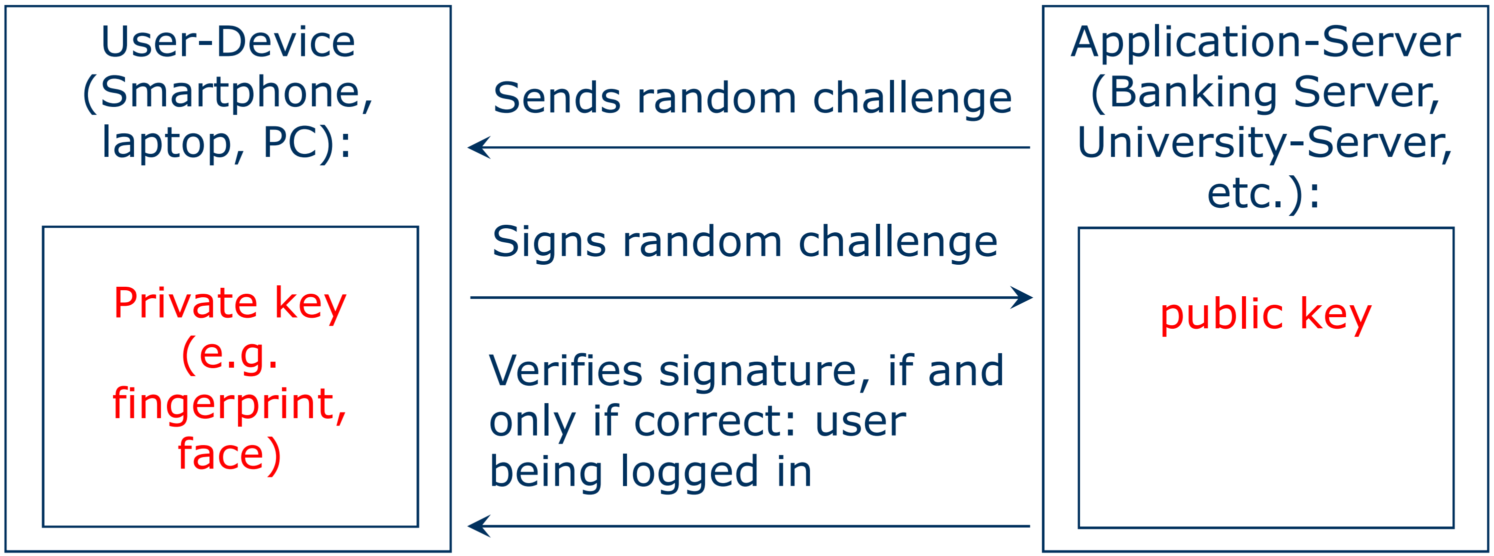
4. FIDO – a short Overview

- FIDO – Fast IDentity Online:
  - non-commercial IT-security organisation
  - founded in February 2013
  - FIDO – kind of ‚base organization‘ for passkeys
- Goal:
  - development of open and non-licensed worldwide authentication methods for the internet

Source: https://de.wikipedia.org/wiki/FIDO-Allianz

5. Passkeys – How they work

Passkey – based *on* ***asymmetric crypto-system***
User creates **keypair**:
Public key – EVERYONE may encrypt a plaintext message
Private key – kept private on local device!
Decryption ONLY possible with private key



- IDEA of asymmetric Crypto systems: Usage of a **one-way- or trapdoor-function**.
- *With public key only* it is impossible to build the inverse function needed for decryption.
- Function can ONLY be inverted with special trick.

Source: https://www.passkeycentral.org/introduction-to-passkeys/how-passkeys-work

6. Example for passkeys – showing the idea behind passkeys …

- RSA – is an *asymmetric crypto-system*
- RSA is based on the *integer factorization problem*, i.e. it is impossible to factor a composite number $n = p \cdot q$ into its prime factors p, q, if p, q are sufficiently large primes
- p and q shall each have at least $512\, bits \approxeq 150\, decimals$ in length!
- RSA encryption: $y = e_{k_{pub\,(e)}}(x) = x^e mod\, n$, with public key: $k_{pub_{(e)}} = (n, e)$ and plain text x
- Since exponentiation is done $modulo\, n$, 2 aspects hold:
  - Exponentiation cannot be inverted (necessary for decryption) by computing the e-th root of y
  - y appears as some large ‚random' number.

Source: C. Paar, J. Pelzl: „Understanding Cryptography", Chapter 7, Springer Verlag, 2010

6. Example for passkeys – showing the idea behind passkeys …

- Problem: How can we calculate plain text x out of cipher text y?

- Side aspect 1: Euler's Phi function $\Phi(n)$: gives the **number** of all elements within $\mathbb{Z}_n = \{0,1,2,\ldots,n-1\}$ that are relatively prime to n

- Side aspect 2: If p, q are primes and $n = p \cdot q$, then $\boldsymbol{\Phi(n) = (p-1) \cdot (q-1)}$.
  - $\Phi(n)$ EASY to calculate, if and only if we know p and q!
  - If we only know n, we CANNOT calculate $\Phi(n)$

- Side aspect 3: if $\gcd(\Phi(n), e) = 1$, i.e. if $\Phi(n)$ and e are relatively prime, there exists an element d (multiplicative inverse of d) within $\mathbb{Z}_n = \{0,1,2,\ldots,n-1\}$, such that:

$$e \cdot d \; mod \; \Phi(n) \equiv 1 \Leftrightarrow \boldsymbol{d = e^{-1} \, mod \, \Phi(n)}$$

Source: C. Paar, J. Pelzl: „Understanding Cryptography", Chapter 7, Springer Verlag, 2010

6. Example for passkeys – showing the idea behind passkeys …

- Side aspect 4: Decyption by 2nd exponentiation $modulo\ n$

$$(\boldsymbol{x^e})^{\boldsymbol{d}} \boldsymbol{mod\ n} = \boldsymbol{x^{e \cdot d}} \boldsymbol{mod\ n} \neq \boldsymbol{x^{e \cdot d\ mod\ n}}\ \boldsymbol{mod\ n}, \text{ instead:}$$

$$(\boldsymbol{x^e})^{\boldsymbol{d}} \boldsymbol{mod\ n} = \boldsymbol{x^{e \cdot d}} \boldsymbol{mod\ n} = \boldsymbol{x}^{\underbrace{\boldsymbol{e \cdot d\ mod\ \Phi(n)}}_{\equiv 1}}\ \boldsymbol{mod\ n}$$

- Conclusion:
  - Decryption of cipher text y not possible by calculating e-th root of y – due to modulo-computation!) BUT
  - Decryption possible by 2nd exponentiation with mulitplicative inverse $\boldsymbol{d = e^{-1}\ mod\ \Phi(n)}$

Source: C. Paar, J. Pelzl: „Understanding Cryptography", Chapter 7, Springer Verlag, 2010

6. Example: RSA - Key Generation and Proof of Correctness

- Suppose smart phone user does the following!
  1. Choose 2 large prime numbers p and q.
  2. Compute $n = p \cdot q$
  3. Compute Euler's Phi function $\Phi(n) = (p-1) \cdot (q-1)$
  4. Select the public exponent $e \in \{1,2, \ldots \Phi(n) - 1\}$ *such that*

$$\gcd(\boldsymbol{\Phi}(\boldsymbol{n}), \boldsymbol{e}) = \mathbf{1}$$

  Publish public key (n,e) – everyone may encrypt a plain text message x

  5. *Compute the private key* d *such that*

$$d \cdot e \equiv 1 \, mod \, \Phi(n) \iff \boldsymbol{d = e^{-1} \, mod \, \Phi(n)}$$

  6. Keep d secret as it is your private key!

Note: Multiplicative inverse $e^{-1} mod \, \Phi(n)$ *may be computed using the extended Euclidean algorithm*

Source: C. Paar, J. Pelzl: „Understanding Cryptography", Chapter 7, Springer Verlag, 2010

6. Example: RSA – real parameters for p, q and n

$$
\begin{aligned}
p = {} & E0DFD2C2A288ACEBC705EFAB30E4447541A8C5A47A37185C5A9 \\
& CB98389CE4DE19199AA3069B404FD98C801568CB9170EB712BF \\
& 10B4955CE9C9DC8CE6855C6123_h \\
q = {} & EBE0FCF21866FD9A9F0D72F7994875A8D92E67AEE4B515136B2 \\
& A778A8048B149828AEA30BD0BA34B977982A3D42168F594CA99 \\
& F3981DDABFAB2369F229640115_h \\
n = {} & CF33188211FDF6052BDBB1A37235E0ABB5978A45C71FD381A91 \\
& AD12FC76DA0544C47568AC83D855D47CA8D8A779579AB72E635 \\
& D0B0AAAC22D28341E998E90F82122A2C06090F43A37E0203C2B \\
& 72E401FD06890EC8EAD4F07E686E906F01B2468AE7B30CBD670 \\
& 255C1FEDE1A2762CF4392C0759499CC0ABECFF008728D9A11ADF_h
\end{aligned}
$$

Source: C. Paar, J. Pelzl: „Understanding Cryptography", Chapter 7, Springer Verlag, 2010

6. Example: RSA – real parameters for e and d

$$
\begin{aligned}
e = {} & 40B028E1E4CCF07537643101FF72444A0BE1D7682F1EDB553E3 \\
& AB4F6DD8293CA1945DB12D796AE9244D60565C2EB692A89B888 \\
& 1D58D278562ED60066DD8211E67315CF89857167206120405B0 \\
& 8B54D10D4EC4ED4253C75FA74098FE3F7FB751FF5121353C554 \\
& 391E114C85B56A9725E9BD5685D6C9C7EED8EE442366353DC39_h \\
d = {} & C21A93EE751A8D4FBFD77285D79D6768C58EBF283743D2889A3 \\
& 95F266C78F4A28E86F545960C2CE01EB8AD5246905163B28D0B \\
& 8BAABB959CC03F4EC499186168AE9ED6D88058898907E61C7CC \\
& CC584D65D801CFE32DFC983707F87F5AA6AE4B9E77B9CE630E2 \\
& C0DF05841B5E4984D059A35D7270D500514891F7B77B804BED81_h
\end{aligned}
$$

Source: C. Paar, J. Pelzl: „Understanding Cryptography“, Chapter 7, Springer Verlag, 2010

6. RSA Example

User (Smartphone)

1. choose $p = 3$ and $q = 11$
2. $n = p \cdot q = 33$
3. $\Phi(n) = (3-1)\,(11-1) = 20$
4. choose $e = 3$ (public exponent)
5. $d \equiv e^{-1} mod\ \Phi(n) \equiv 7\ mod\ 20$

$$x = y^d mod\ n$$
$$\equiv 31^7 mod\ 33 \equiv 4$$

Note: prime factorization of n is not possible (only known by user!)
→ $\Phi(n)$ can ONLY be calculated by user!!

1. Publish
$K_{pub} = (n = 33, e = 3)$

$\longrightarrow$

2. Send:
Cipher text: $y = 31)$

$\longleftarrow$

3. Return plain text
$x = 4$

$\longrightarrow$

Application Server
(e.g. Banking Server)

Assume: random plain text $x = 4$
Server calculates cipher text:
$$y = x^e mod\ n = 4^3 mod\ 33 = 31$$

Server verifies that returned plain text x is correct → Login o.k.

Source: C. Paar, J. Pelzl: „Understanding Cryptography", Chapter 7, Springer Verlag, 2010

7. Passkey Benefits

- Passkey sign-in:
  - more convenient than passwords, since no passwords need to be remembered and typed by user
  - always strong (as compared to – possibly – weak passwords!)
  - greatly increases security, since passkeys are phishing-resistant
  - up to 20% higher success rate than typing passwords
  - up to 75% faster than typing passwords
  - more successful, faster and more secure authentication!
- Credential not used across contexts (like single password for various accounts).
- Large global service providers like Amazon, CVS Health, Google, Nintendo, Intuit, and many others offer FIDO-based sign-in with passkeys.
- passkey sign-in means:
  - Better service delivery
  - More transaction completion
  - Less account recovery events
  - Less breach risk

Source: https://www.passkeycentral.org/introduction-to-passkeys/

8. Conclusion

- In some way history repeats:
  - Symmetric cryptography known and used much longer than asymmetric cryptography
  - Used for long time: Passwords ⇔ symmetric crypto system while
  - New system: Passkeys ⇔ asymmetric crypto system

- Passkeys
  - Are always strong
  - Do not need to be remembered (as passwords do)
  - Are phishing resistant
  - Provide a much higher security than passwords
  - Provide fast and reliable performance

- Passkeys are an elegant and sophisticated method for secure web- and application access!

Thank you !

Any questions?