

2BPR



BLIND PASSWORD REGISTRATION FOR TWO-SERVER PASSWORD AUTHENTICATED
KEY EXCHANGE AND SECRET SHARING PROTOCOLS

Franziskus Kiefer & Mark Manulis

Gliederung

- Hintergrund
- Motivation
- Begriffe
- Protokoll
- Sicherheitsanalyse
- Fazit



Hintergrund - Multiusersystem



Wie sichert man Multi-User-Systeme gegen Missbrauch ab?



- Lange Passwörter
- Sonderzeichen & Zahlen
- Regelmäßiges Ändern



- Kontrolle der Richtlinien
- Plain Passwort Datenbanken
- Passworthashes nicht sicher

Wie sichert man Multi-User-Systeme gegen Missbrauch ab?



Individueller Hash durch Salt
Verhindert Lookuptables
Regelmäßiges Ändern



Zu kurzer oder schlechter Salt
Kontrolle der Richtlinien

Wie sichert man Multi-User-Systeme gegen Missbrauch ab?



Mehrere Server nutzen
Passwort „verteilen“
2PAKE oder 2PASS



Kontrolle der Richtlinien

Hintergrund – 2PAKE & 2PASS

PASSWORD AUTHENTICATED KEY EXCHANGE

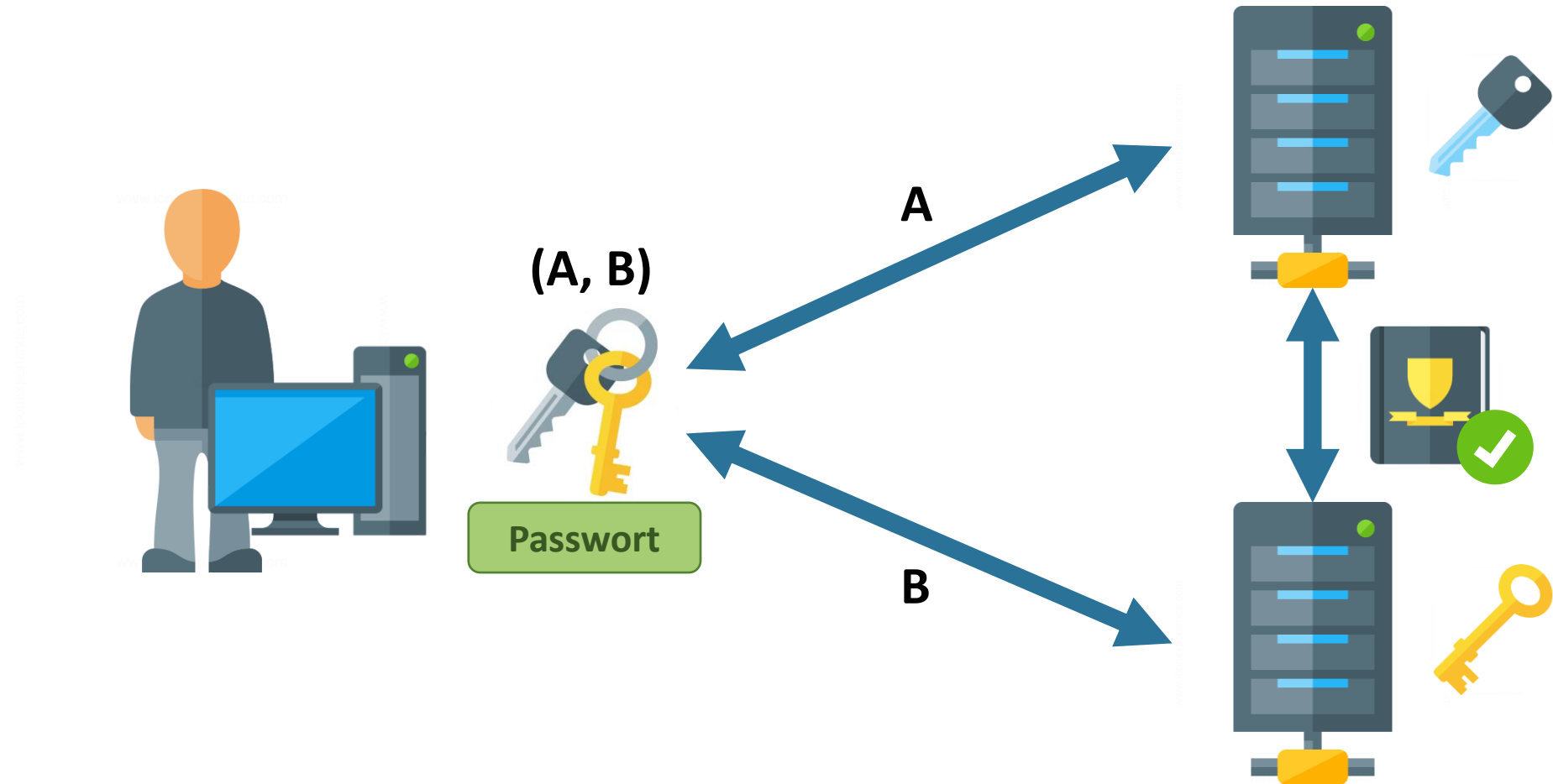
- Passwort wird in s_1 und s_2 geteilt
- s_1 und s_2 auf zwei Servern speichern
- Zusammenarbeit der Server bei Login
- Kein Server kennt das ganze Passwort

PASSWORD AUTHENTICATED SECRET SHARING

- Passwort mit hoher Entropie auf mehreren Servern verteilen
- Passwort mit niedriger Entropie autorisiert den Abrufprozess des ganzen Passworts

+ Man kann beiden Servern zu 100% vertrauen

Hintergrund – 2PAKE & 2PASS



Hintergrund – 2PAKE & 2PASS



Man in the Middle



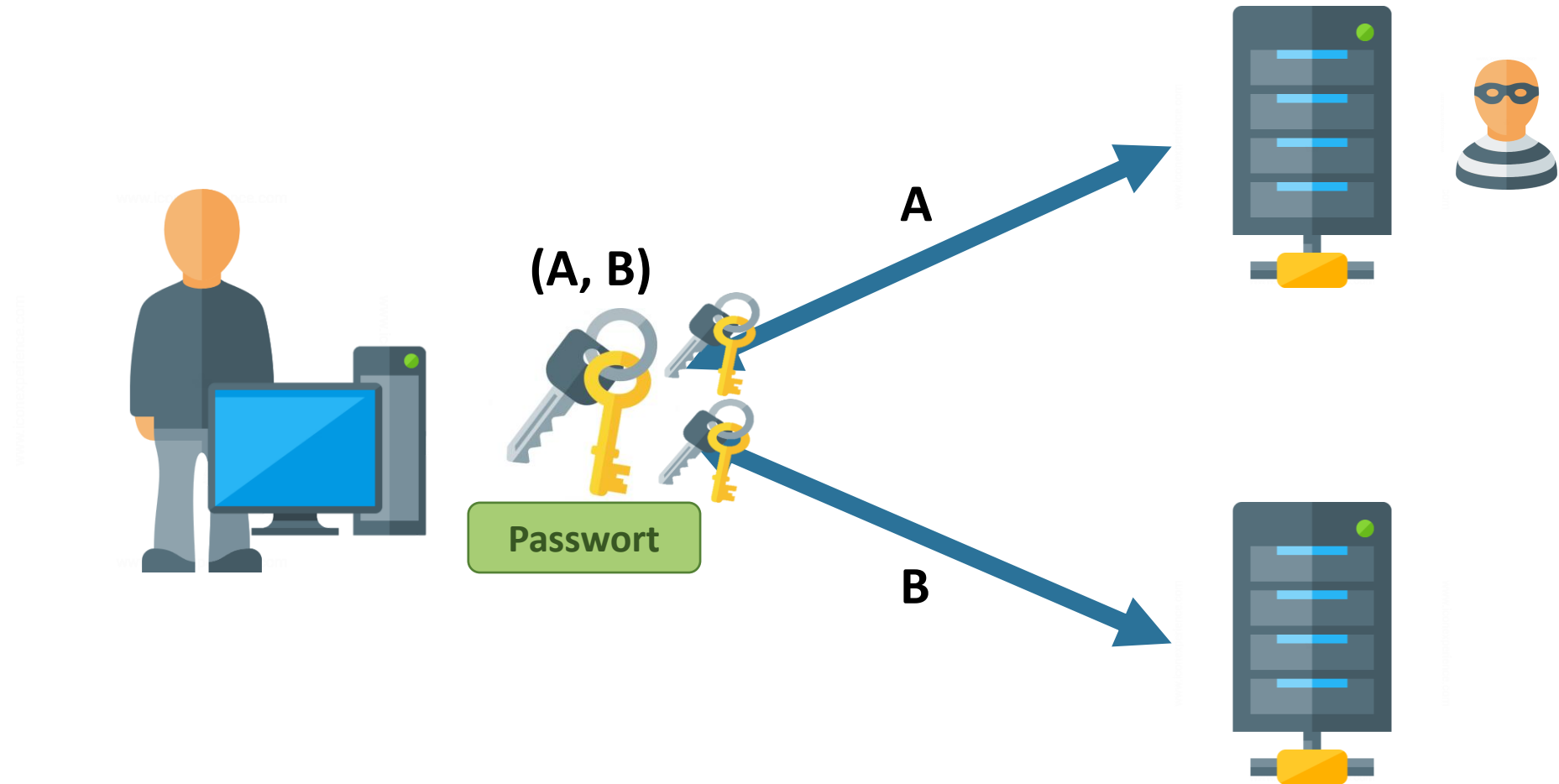
Brutforce

Gliederung

- Hintergrund
- **Motivation**
- Begriffe
- Protokoll
- Sicherheitsanalyse
- Fazit



Motivation



Motivation



2BPR



~~Kontrolle der Richtlinien~~

- Commitments & Zero Knowledge Password Policy Checks (ZKPPC)
 - Keine offline Wörterbuch Attacken möglich
 - Sichere Registrierung von neuen Passwörtern
- ➔ Sicherer Registrierungsprozess in 2PAKE & 2PASS Multiusersystemen

Gliederung

- Hintergrund
- Motivation
- **Begriffe**
- Protokoll
- Sicherheitsanalyse
- Fazit



Commitment

- Szenario: SSP oder Münzwurf über Internet spielen
- Bedingung: Kein TrustCenter vorhanden



- ⚡ Keiner der Spieler darf auf den Zug des anderen reagieren
- ⚡ Das Festlegen auf Schere / Stein/ Papier muss verbindlich sein
- 💡 Verwendung eines Commitments

Commitment

- Binding



Bob legt sich auf Zahl fest → kein Umentscheiden möglich



Bob wählt Stein, Alice wählt Papier → kein Umentscheiden möglich

- Hiding

Gleichzeitiges abgeben der Commitments ist nicht möglich

Der Inhalt muss bis zum „Aufdecken“ versteckt bleiben

Pedersen Commitment

Basis: Diskreter Logarithmus

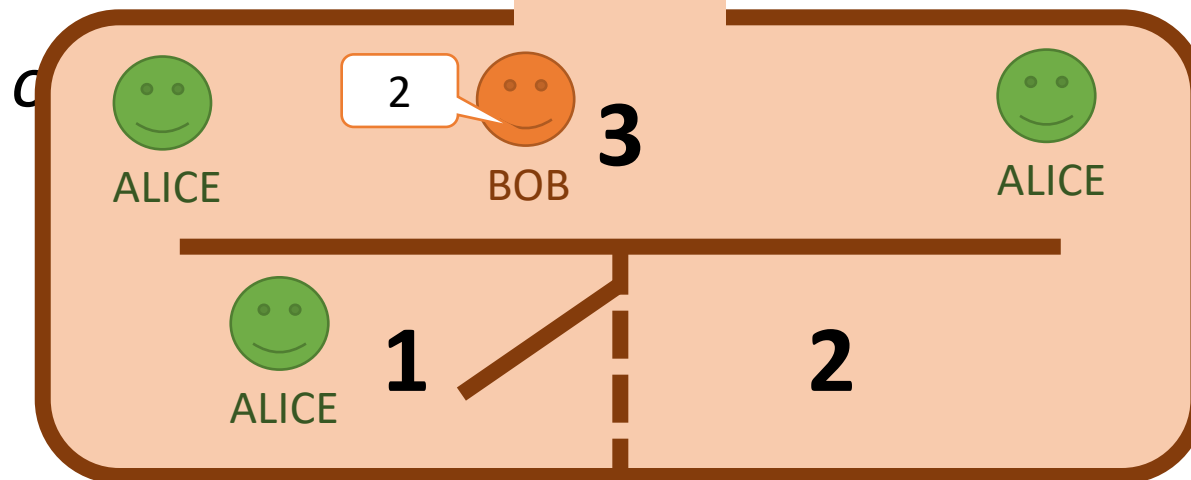
TRAPDOOR COMMITMENT

1. Setup Bob legt Primzahlen p, q, g und v fest
2. Committed Alice berechnet Commitment aus $c = g^r v^m$
3. Aufdecken Alice sendet r und m an Bob





- + Unconditional hiding
- + Computational binding
- + Großer Wertebereich für Nachricht
- + Additiv homomorph

Zero Knowledge Proof

„Beweiser P überzeugt Verifizierer V davon, dass er ein Geheimnis kennt



Informationen zu offenbaren.“^[6]

- 4  ALICE  Kennt kein Geheimnis
-  Zu 50% falsche Seite
-  Kennt Geheimnis
- Sicher zu $P = 1 - 2^{-n}$

Zero Knowledge Proof

- Vollständigkeit

Ist x ein Element der Sprache L , dann soll V fast immer akzeptieren.

- Zuverlässigkeit

Ist x kein Element der Sprache L , also ist P unehrlich, soll V fast immer ablehnen.

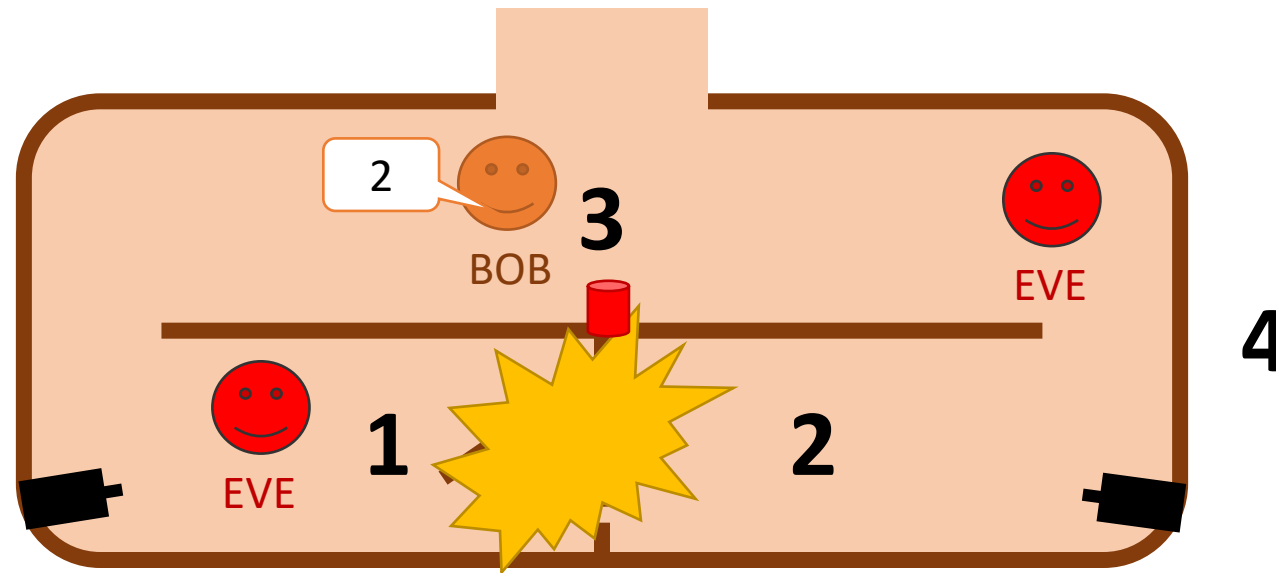
- Zero-Knowledge-Eigenschaft

Es darf nur Wissen über die Gültigkeit einer zu beweisenden Aussage gewonnen werden. Ein dritter, der das Verfahren beobachtet gewinnt keine Informationen.

Zero Knowledge Proof of Knowledge

- Zuverlässigkeit

Es gibt einen Extraktor *Ext*, der den korrekten Beweis aus einem bösen *P* extrahieren kann, sodass *V* den Beweis doch noch ablehnt.



Passwörter

- Passwort Richtlinien

Besteht aus regulärem Ausdruck & Angabe für die Mindestlänge des Passworts

Beispiel: $f = f(R, n) = (ulldds, 10)$

Um die beiden Richtlinien zu kombinieren wird $f = f_1 \cap f_2$ gebildet

Beispiel: $f1 \cap f2 = (max(R_1, R_2), max(n_1, n_2)) \rightarrow$ Mutual Password Policy

- Passwörter werden in Integer umgewandelt

$$\pi = PWDtoINT(pw)$$

Passwörter

- Passwörter können zerteilt werden (Password Sharing)

$$\pi = s_1 + s_2$$

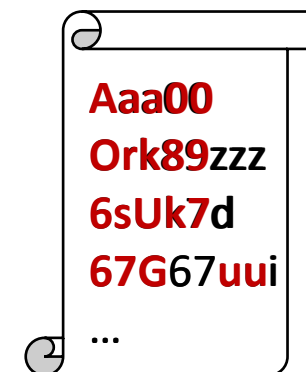
s_1 kann auf Server 1 hinterlegt werden

s_2 kann auf Server 2 hinterlegt werden

- Passwort Wörterbuch

Liste aller richtlinienkonformen Passwörter

Beispiel: $f = (ulldd, 5)$



SIGNIFIKANT

Gliederung

- Hintergrund
- Motivation
- Begriffe
- **Protokoll**
- Sicherheitsanalyse
- Fazit



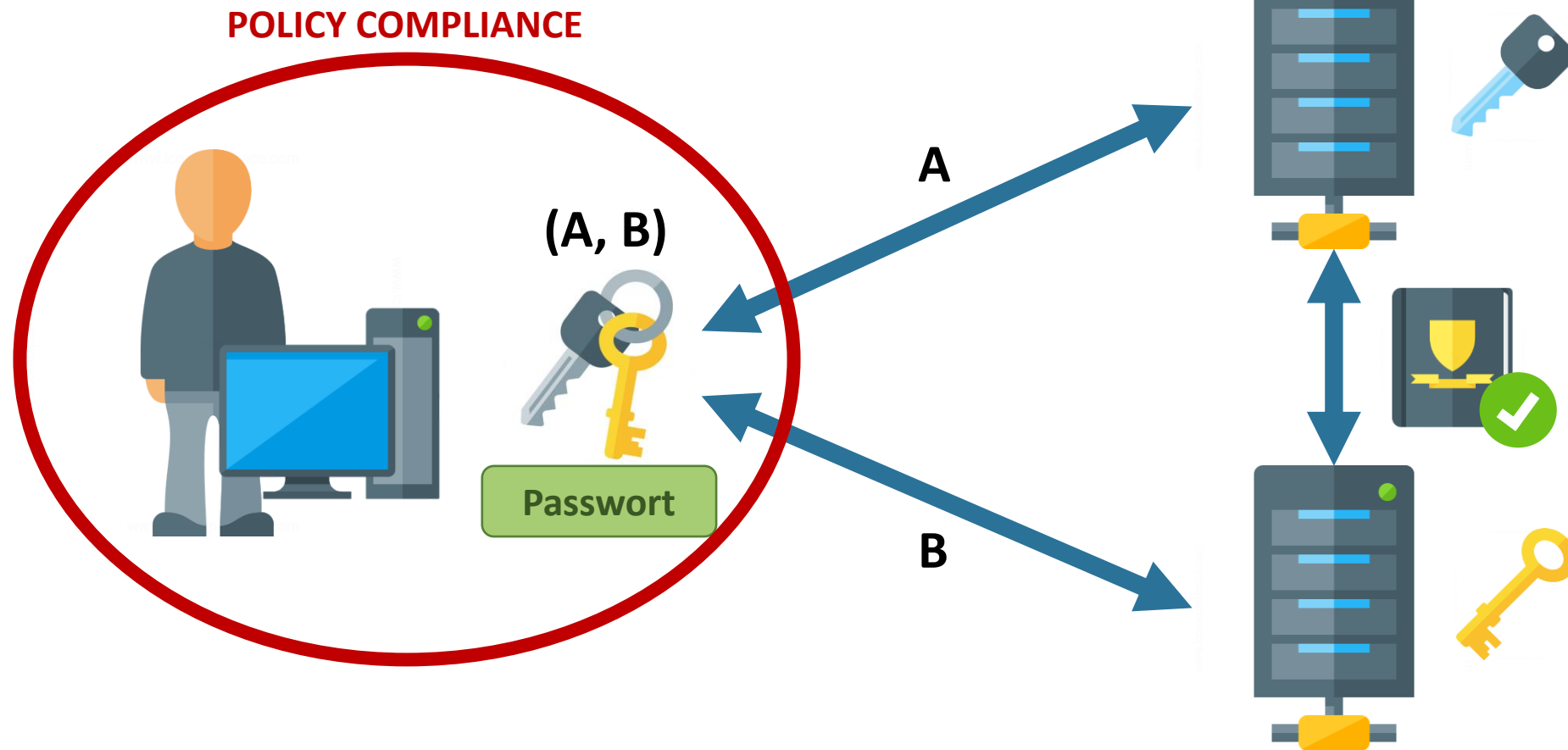
2BPR – Sicherheitsmodell

1. POLICY COMPLIANCE

Die beiden ehrlichen Server akzeptieren ihren Passwortshare, wenn dieser Policy konform ist, ansonsten lehnen sie den Share ab.

→ Wenn beide den Share akzeptieren ist das Passwort konform zur Mutual Password Policy.

2BPR – Sicherheitsmodell



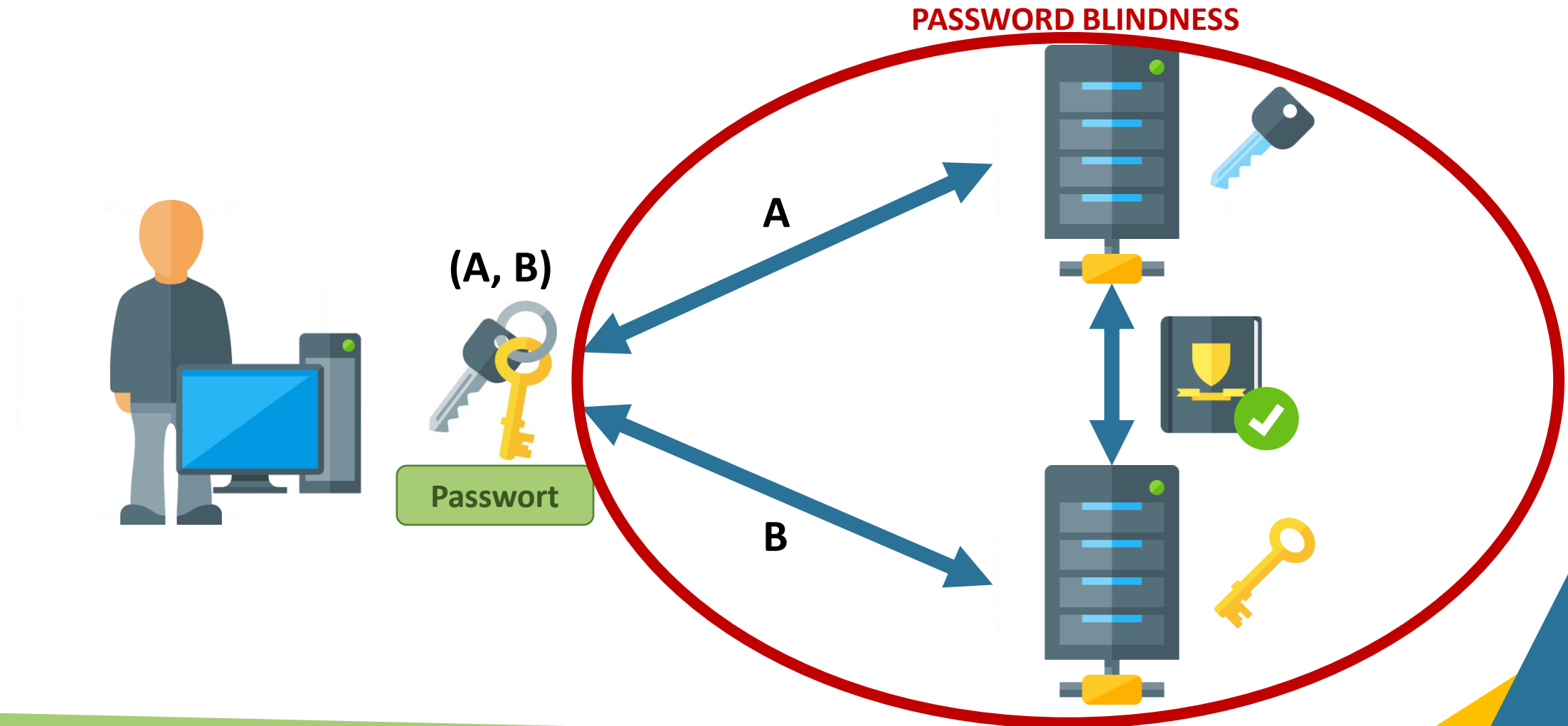
2BPR – Sicherheitsmodell

2. PASSWORD BLINDNESS

Ein korumpierter Server soll nur erfahren, ob das Passwort Policy konform ist. Weitere Infos über das Passwort bleiben geheim.

→ Offline Wörterbuch Attacken sind dadurch zwecklos solange ein Server ehrlich bleibt.

2BPR – Sicherheitsmodell



2BPR – Phasen

1. Client Vorbereitung

Der Client bereitet Primzahlen, Passwort und Commitments vor

2. Passwort Registrierung

Der Client bestätigt die Konformität des Passworts gegenüber den Servern

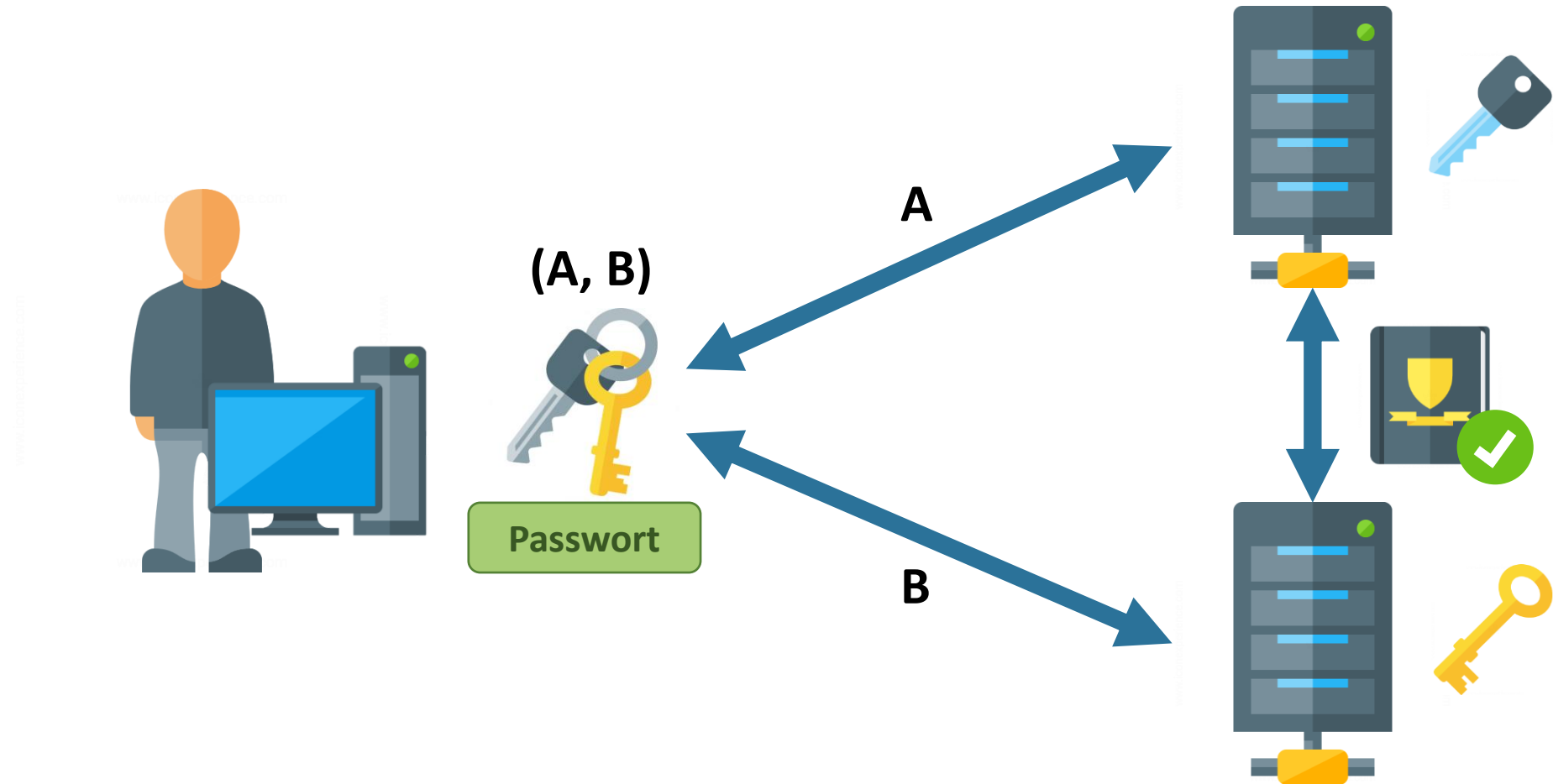
3. Share Verifikation

Die Server testen ob der Client beiden Servern das selbe Passwort mitgeteilt hat

2BPR – Client Vorbereitung

- Erklärung zur Vorbereitung
- Mit Anschließender graphischer Darstellung

2BPR – Client Vorbereitung



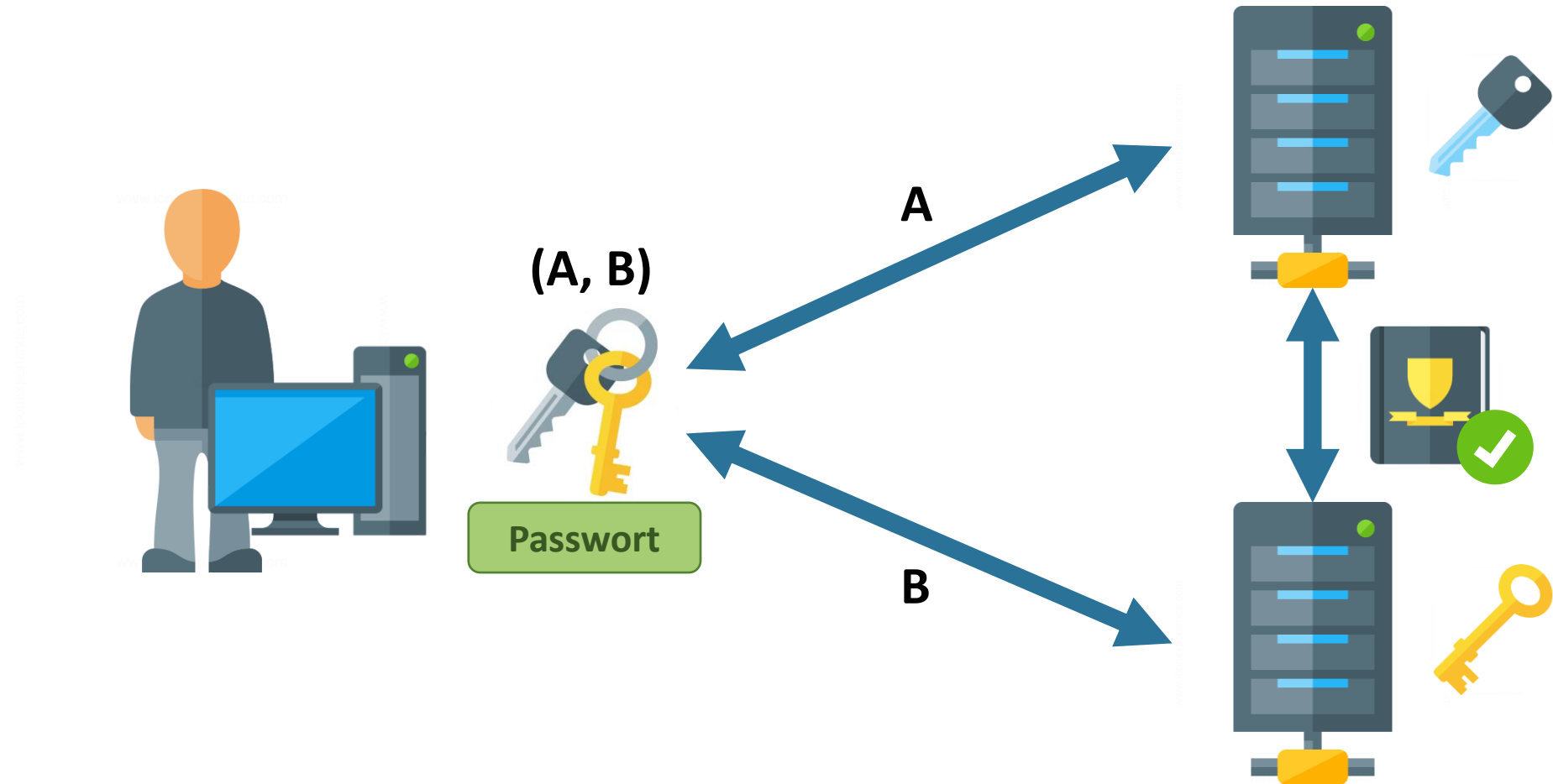
2BPR – Passwort Registrierung

- Erklärung zur Passwort Registrierung
- Mit Anschließender graphischer Darstellung

2BPR – Passwort Registrierung

- Erklärung zu
 - Proof of Membership
 - Proof of Shuffle
 - Proof of Correctness

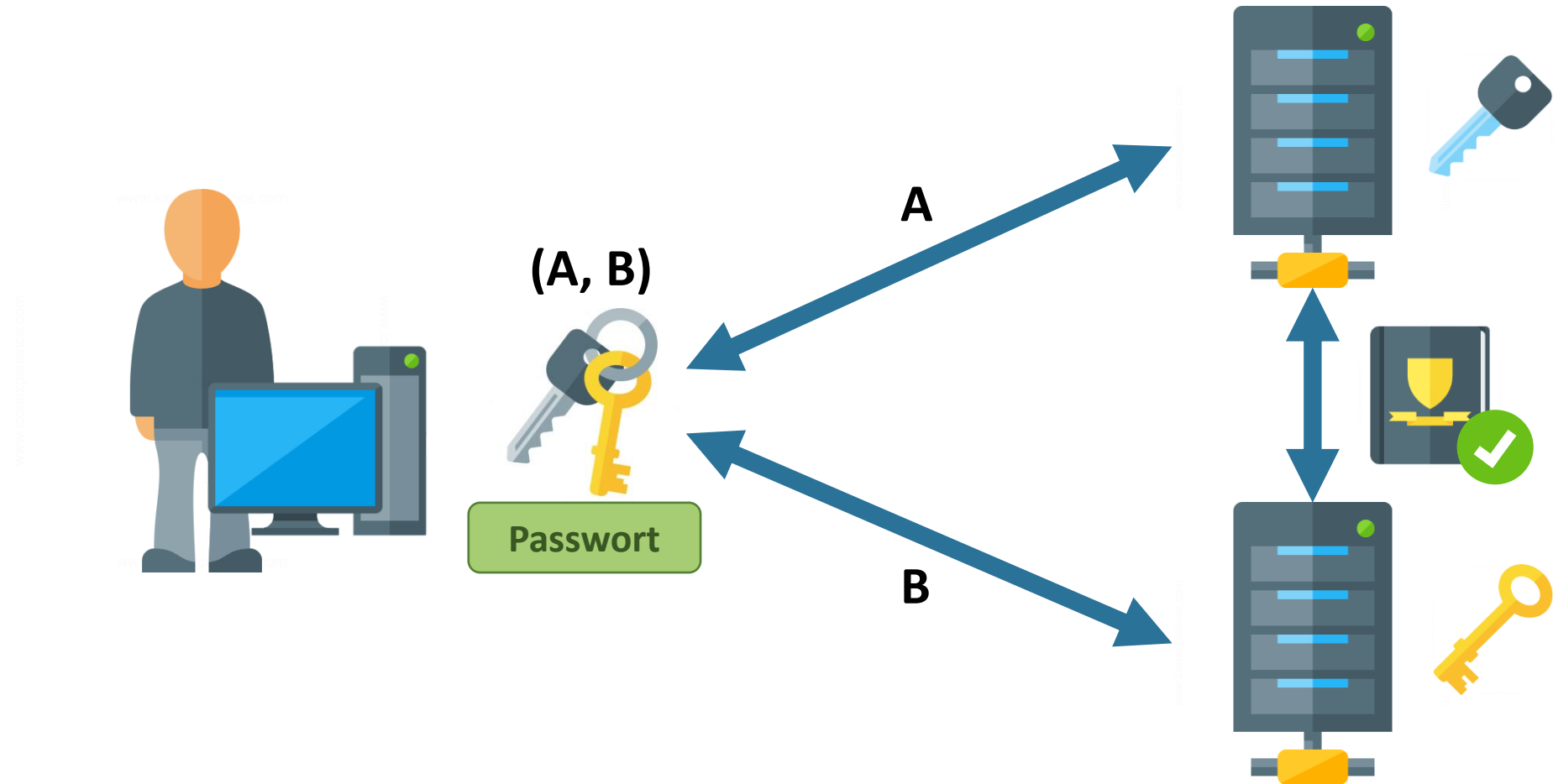
2BPR – Passwort Registrierung



2BPR – Share Verifikation

- Erklärung zur Share Verifikation
- Mit Anschließender graphischer Darstellung

2BPR – Share Verifikation



Gliederung

- Hintergrund
- Motivation
- Begriffe
- Protokoll
- Sicherheitsanalyse
- Fazit



- Spielebasierte Sicherheitsanalyse aus dem Paper

Gliederung

- Hintergrund
- Motivation
- Begriffe
- Protokoll
- Sicherheitsanalyse
- Fazit



Fazit – Performance

- Performance der Algorithmen
- Python Beispiel

Fazit – Anwendung 2PAKE/2PASS

- Probleme bei der Anwendung bei 2PAKE/2PASS
- Auf was muss man achten?

Fazit

- Eigenes Fazit zu Paper
- Eigenes Fazit zu 2BPR / 2PAKE / 2PASS
- Fazit zu Commitments und Zero Knowledge Proofs

Bildquellen

- <http://wfarm2.dataknet.com/static/resources/icons/set112/8cbf6bf1.png> 23.11.2017 14:38
- <https://www.iconexperience.com> 24.11.2017 11:01
- https://upload.wikimedia.org/wikipedia/commons/thumb/4/45/New_Logo_Gmail.svg/1200px-New_Logo_Gmail.svg.png 23.12.2017 13:57
- [https://fthmb.tqn.com/jRaoLvoOhFQWEWmMmyiZRcL_NHg=/768x0/filters:no_upscale\(\)/Outlook-icon-57f005363df78c690f62c7af.png](https://fthmb.tqn.com/jRaoLvoOhFQWEWmMmyiZRcL_NHg=/768x0/filters:no_upscale()/Outlook-icon-57f005363df78c690f62c7af.png) 23.12.2017 13:57
- https://lh3.googleusercontent.com/UrY7BAZ-XfXGpfkeWg0zCCeo-7ras4DCoRaIC_WXXWTK9q5b0lw7B0YQMsVxZaNB7DM=w300 23.12.2017 13:57
- <https://lh3.googleusercontent.com/dSDutSmwU9LMJDCs9PaJI1JjXQthi8IDNRHPvil1NzocGTwuWC-PTAF6QiaGtCgF0A=w300> 23.12.2017 13:57
- https://upload.wikimedia.org/wikipedia/commons/thumb/1/18/GitLab_Logo.svg/1200px-GitLab_Logo.svg.png 23.12.2017 13:57
- https://assets-cdn.github.com/images/modules/open_graph/github-mark.png 23.12.2017 13:57
- <https://lh3.googleusercontent.com/z7oKSvTI-2ynS5bHggIctR9GVkS8sGKqpDlfCvgxLo0du7Az00u6XpJ0LLyvzBusW-Jd=w300> 23.12.2017 13:57
- https://lh3.googleusercontent.com/Dq-mZ5mmdE6aFPeD61DNIVTwYSI75UwHBYDq_BxBZOMSzCBnQ5OCC4-LjfP42tDlyw=w300 23.12.2017 13:57

Bildquellen

- <http://www.horizont.net/news/media/2/Web-hat-es-nic-gescha-Unddu-zu-ein-erfolgreic-Por--16438.jpeg> 23.12.2017 14:15
- https://logos-download.com/wp-content/uploads/2016/10/GMX_logo_blue.png 23.12.2017 14:16
- <https://tradingeducationblogs.com/wp-content/uploads/2017/03/snapchat-logo.png> 23.12.2017 16:15
- https://d1x0mwiac2rqwt.cloudfront.net/bab0a0c4b1c3135a24bd0518417b66e3/as/logo_todoist_schema.png 23.12.2017 13:57
- https://upload.wikimedia.org/wikipedia/de/thumb/9/9f/Twitter_bird_logo_2012.svg/1200px-Twitter_bird_logo_2012.svg.png 23.12.2017 13:57
- https://www.facebook.com/images/fb_icon_325x325.png 23.12.2017 13:57
- https://pixabay.com/p-1581266/?no_redirect 23.12.2017 14:08
- <https://upload.wikimedia.org/wikipedia/commons/thumb/8/83/Sparkasse.svg/2000px-Sparkasse.svg.png> 23.12.2017 14:08
- https://upload.wikimedia.org/wikipedia/commons/thumb/a/ab/Volksbank_Logo.svg/1000px-Volksbank_Logo.svg.png 23.12.2017 14:08
- <http://millionmedia.com/wp-content/uploads/2014/11/deezer-logo-circle.png> 23.12.2017 14:11
- <http://logodatabases.com/wp-content/uploads/2012/03/deutsche-bank.jpg> 23.12.2017 14:11

Internetquellen

- [1] <http://www.itwissen.info/Mehrbenutzersystem-multi-user-system.html> 23.12.2017 15:07
- [2] <https://arstechnica.com/information-technology/2013/11/> 24.11.2017 09:38
- [3] <https://techcrunch.com/2009/12/14/rockyou-hack> 24.11.2017 09:44
- [4] <https://www.reuters.com/article/us-adobe-cyberattack/> 24.11.2017 09:50
- [5] <https://crackstation.net/hashing-security.htm> 24.11.2017 08:17
- [6] <https://de.wikipedia.org/wiki/Zero-Knowledge-Beweis> 30.12.2017 19:41
- https://en.wikipedia.org/wiki/Password-authenticated_key_agreement 24.11.2017 10:12
- <http://ieeexplore.ieee.org/document/7450662/> 24.11.2017 10:23
- <https://budickda.gitbooks.io/commitment-schemes/content/chapter3.html> 26.12.2017 16:15