

Identification and Authentication

Daniel Bosk

Department of Information and Communication Systems,
Mid Sweden University, Sundsvall

24th April 2017

1 Introduction

■ Identification and Authentication

2 Bootstrapping Authentication

- What is bootstrapping?
- Problems with Bootstrapping
- Single Sign-On

3 Authenticating

- User-machine authentication
- Multi-factor user-authentication
- Time of check, time of use
- Machine-user authentication

4 Securing Authentication

- Guessing Passwords
- The Password File
- Alternative Approaches
- Anonymous Credentials

Definition (Identifier)

- An identifier is a piece of data that uniquely identifies some entity.

Example (Identifiers)

- An email address identifies a user uniquely in the email system.
- A username identifies a user in some system.
- A passport number uniquely identifies a passport issued by a country.

Definition (Authentication)

- Some entity claims an identity: “identifier X identifies me”.
- Authentication is about verification.
- That entity must *convince* us that its claim is true.

Exercise: How can we authenticate

- the claim of an email address?
- the claim of a username in some system?
- the claim of a passport number?
- the claim of a national identity in some country?

Definition (Authentication)

- Some entity claims an identity: “identifier X identifies me”.
- Authentication is about verification.
- That entity must *convince* us that its claim is true.

Exercise: How can we authenticate

- the claim of an email address?
- the claim of a username in some system?
- the claim of a passport number?
- the claim of a national identity in some country?

Example (User authentication)

Identification First you enter your username to *identify* yourself.

Authentication Then you enter your password to *authenticate* that you are truly you.

Exercise

Why does this work?

Example (User authentication)

Identification First you enter your username to *identify* yourself.

Authentication Then you enter your password to *authenticate* that you are truly you.

Exercise

Why does this work?

- 1 Introduction
 - Identification and Authentication
- 2 Bootstrapping Authentication
 - What is bootstrapping?
 - Problems with Bootstrapping
 - Single Sign-On
- 3 Authenticating
 - User-machine authentication
 - Multi-factor user-authentication
 - Time of check, time of use
 - Machine-user authentication
- 4 Securing Authentication
 - Guessing Passwords
 - The Password File
 - Alternative Approaches
 - Anonymous Credentials

What is bootstrapping?

Bootstrapping: A hen-and-egg problem

- Alice is not registered in our authentication system.
- We want to register her as a user in our system.
- How do we know Alice is actually Alice?
- Since she's not registered we cannot authenticate her.

Exercise

- Any quick workarounds that comes to mind?
- When is this a problem and when is it not?

What is bootstrapping?

Bootstrapping: A hen-and-egg problem

- Alice is not registered in our authentication system.
- We want to register her as a user in our system.
- How do we know Alice is actually Alice?
- Since she's not registered we cannot authenticate her.

Exercise

- Any quick workarounds that comes to mind?
- When is this a problem and when is it not?

What is bootstrapping?

Bootstrapping: A hen-and-egg problem

- Alice is not registered in our authentication system.
- We want to register her as a user in our system.
- How do we know Alice is actually Alice?
- Since she's not registered we cannot authenticate her.

Exercise

- Any quick workarounds that comes to mind?
- When is this a problem and when is it not?

What is bootstrapping?

Solution (We don't care who Alice is)

- *We simply set up authentication when Alice creates the account.*
- *Now we can authenticate whoever set up the account.*

Example

This is the solution used by most web services.

What is bootstrapping?

Solution (We don't care who Alice is)

- *We simply set up authentication when Alice creates the account.*
- *Now we can authenticate whoever set up the account.*

Example

This is the solution used by most web services.

What is bootstrapping?

Solution (We care who Alice actually is)

- *We can require ID checks etc. to set up the authentication mechanisms using a helpdesk.*
- *If we have address etc., then we can send the credentials via mail (be it snailmail or email).*

Example (The university account)

This is how your university account was set up.

What is bootstrapping?

Solution (We care who Alice actually is)

- *We can require ID checks etc. to set up the authentication mechanisms using a helpdesk.*
- *If we have address etc., then we can send the credentials via mail (be it snailmail or email).*

Example (The university account)

This is how your university account was set up.

What is bootstrapping?

Solution (We care who Alice actually is)

- *We can require ID checks etc. to set up the authentication mechanisms using a helpdesk.*
- *If we have address etc., then we can send the credentials via mail (be it snailmail or email).*

Example (The university account)

This is how your university account was set up.

What is bootstrapping?

Exercise

- How is Alice authenticated when she applies for an ID?

What is bootstrapping?

Example (Signal, WhatsApp, ...)

- The identity is a (mobile) phone number.
- Send a text message with a code.

Remark

- Phone provider can impersonate.
- Government can impersonate (forcing phone provider).

What is bootstrapping?

Example (Signal, WhatsApp, ...)

- The identity is a (mobile) phone number.
- Send a text message with a code.

Remark

- Phone provider can impersonate.
- Government can impersonate (forcing phone provider).

- 1 Attacker *intercepts* a password on account creation.
 - User starts bootstrapping.
 - Password is sent to user.
 - Attacker grabs password.
- 2 Attacker *impersonates* the legitimate user.
 - Attacker starts bootstrapping.
 - User remains unaware.
 - Service cannot distinguish attacker and user.

- 1 Attacker *intercepts* a password on account creation.
 - User starts bootstrapping.
 - Password is sent to user.
 - Attacker grabs password.
- 2 Attacker *impersonates* the legitimate user.
 - Attacker starts bootstrapping.
 - User remains unaware.
 - Service cannot distinguish attacker and user.

- It can be costly to manage.
- Sometimes it is a continuous process, if the same bootstrapping procedure is also used for *recovery from failure*.
- Make sure the system can handle forgotten, lost or aged authentication means.

- We could let someone else who has solved the problem already do the authentication for us.
 - This way the user only needs one username and password, and he or she only needs to sign in once.
 - However, this makes the SSO provider a very attractive target.
 - And they are forced to solve our problem anyway.
 - The problem is, now we need to trust them to do it properly
- ...

Example (We don't care who Alice is)

We can use Google, Facebook etc.

Example (We care who Alice is)

We can use e.g. BankID.

Remark

The SSO-service must have done bootstrapping as rigorously as we would have.

- 1 Introduction
 - Identification and Authentication
- 2 Bootstrapping Authentication
 - What is bootstrapping?
 - Problems with Bootstrapping
 - Single Sign-On
- 3 Authenticating
 - User-machine authentication
 - Multi-factor user-authentication
 - Time of check, time of use
 - Machine-user authentication
- 4 Securing Authentication
 - Guessing Passwords
 - The Password File
 - Alternative Approaches
 - Anonymous Credentials

Example (Identifiers)

- Username or User ID
- The person who opened the account
- Personal Identification Number (Swe. personnummer, Eng. Social Security Number)
- Fingerprint
- Iris scan
- DNA sequence . . .

Exercise

Any other methods of user identification that you have encountered?

Example (Identifiers)

- Username or User ID
- The person who opened the account
- Personal Identification Number (Swe. personnummer, Eng. Social Security Number)
- Fingerprint
- Iris scan
- DNA sequence . . .

Exercise

Any other methods of user identification that you have encountered?

Example (Identifiers)

- Username or User ID
- The person who opened the account
- Personal Identification Number (Swe. personnummer, Eng. Social Security Number)
- Fingerprint
- Iris scan
- DNA sequence . . .

Exercise

Any other methods of user identification that you have encountered?

Example (Methods for user authentication)

- Something you *know*
- Something you *have*
- *Where* you are
- *Who* you are
- What you *do*

Example (Methods for user authentication)

- Something you *know*
- Something you *have*
- *Where* you are
- *Who* you are
- *What* you *do*

Example (Methods for user authentication)

- Something you *know*
- Something you *have*
- *Where* you are
- *Who* you are
- *What* you *do*

Example (Methods for user authentication)

- Something you *know*
- Something you *have*
- *Where* you are
- *Who* you are
- What you *do*

Definition (Multi-factor authentication)

- Combine two or more methods of authentication.

Example (Methods for user authentication)

- Something you *know*
- Something you *have*
- *Where* you are
- *Who* you are
- What you *do*

Example (Single-factor authentication)

Identification Username or similar

Authentication Something you know, i.e. a password

Example (Multi-factor authentication)

Identification Username or similar

Authentication Something you know together with something you have, e.g. password and mobile phone

Example (Single-factor authentication)

Identification Username or similar

Authentication Something you know, i.e. a password

Example (Multi-factor authentication)

Identification Username or similar

Authentication Something you know together with something you have, e.g. password and mobile phone

Exercise

- Whenever we authenticate a user, we do this for a purpose.
- When does this authentication take place in relation to when we make use of it?

Example

- Usually we authenticate a user in the beginning of a session, e.g. at login.
- Equally often we assume the user is authenticated during the entire session, even when fetching coffee, going by the printer – or even when out to lunch.
- Who knows what happens when the user is away from the computer, one thing is for sure: the computer will not know the difference!

Solution

- *This problem can be solved with repeated authentication.*
- *We could lock our system, either manually or by time-out.*
- *We could also authenticate anew when we need to do something requiring more privileges, and if it has been a while since last time – compare with `sudo(8)`.*

Remark

- What we actually need is *continuous* authentication.

Solution

- *This problem can be solved with repeated authentication.*
- *We could lock our system, either manually or by time-out.*
- *We could also authenticate anew when we need to do something requiring more privileges, and if it has been a while since last time – compare with `sudo(8)`.*

Remark

- What we actually need is *continuous* authentication.

Remark

- The issue we have solved so far is to design means for the system to identify and authenticate different users.
- We have another important problem to solve too, how does the user know it is the system he or she is authenticating him- or herself to?
- Thus enters the problem of spoofing, phishing, and social engineering ...

Definition (Spoofing/Masquerading)

- Attacker masquerades as authorized.
- To a system: impersonates authorized user.
- To a user: impersonates authorized system/UI.

Definition (Phishing)

- A masquerading attack trying to collect sensitive data.
- E.g. email from IT department requesting the password.

Definition (Social Engineering)

- The general class of attacks on humans.
- Exploits fallacies in human psychology.
- Parent category of phishing.
- Can be very advanced.

Definition (Phishing)

- A masquerading attack trying to collect sensitive data.
- E.g. email from IT department requesting the password.

Definition (Social Engineering)

- The general class of attacks on humans.
- Exploits fallacies in human psychology.
- Parent category of phishing.
- Can be very advanced.

Exercise

How can we prevent spoofed interfaces?

Example

- Show the user the number of failed login attempts.
- Show the time and location for the last successful login.
- This allows for *detection*.

Example

- We also have the trusted path.
- E.g. Windows uses the Ctrl+Alt+Del to bring up the authentication dialogue upon login.
- This allows for *prevention*.

Remark: Problem with social engineering

- These are attacks on higher levels, e.g. an email or phone call.
- Difficult to check algorithmically.

Example

- Phone call to helpdesk from a “user” in need.
- Stressful situation, willingness to help, ...

Example (Solution?)

- Authenticated phone calls.
- E.g. display caller ID clearly.

“Machine–user authentication: from the user to the machine”

Remark: Problem with social engineering

- These are attacks on higher levels, e.g. an email or phone call.
- Difficult to check algorithmically.

Example

- Phone call to helpdesk from a “user” in need.
- Stressful situation, willingness to help, ...

Example (Solution?)

- Authenticated phone calls.
- E.g. display caller ID clearly.

“M... .. G... .. L... .. L... ..”

Solution

- *Educate and train users to spot these attempts.*
- *Keep strong policies for recovering from authentication failures.*
- *Technological tools and good practices can support users.*

- 1 Introduction
 - Identification and Authentication
- 2 Bootstrapping Authentication
 - What is bootstrapping?
 - Problems with Bootstrapping
 - Single Sign-On
- 3 Authenticating
 - User-machine authentication
 - Multi-factor user-authentication
 - Time of check, time of use
 - Machine-user authentication
- 4 Securing Authentication
 - Guessing Passwords
 - The Password File
 - Alternative Approaches
 - Anonymous Credentials

Guessing Passwords

- Guessing passwords is like searching for a needle in a haystack.
- (Un)fortunately, the needle is placed by a human — not uniformly randomly!
- This makes guessing easier.
- Human-chosen passwords will only occupy parts of the password space.

Guessing Passwords

- Guessing passwords is like searching for a needle in a haystack.
- (Un)fortunately, the needle is placed by a human — not uniformly randomly!
- This makes guessing easier.
- Human-chosen passwords will only occupy parts of the password space.

Guessing Passwords

- The effort is a spectrum.
- It ranges from brute-force exhaustive search ...
- ... via “educated guessing” ...
- ... to getting the password from the user directly.

Guessing Passwords

- The effort is a spectrum.
- It ranges from brute-force exhaustive search ...
- ... via “educated guessing” ...
- ... to getting the password from the user directly.

Guessing Passwords

- The effort is a spectrum.
- It ranges from brute-force exhaustive search ...
- ... via “educated guessing” ...
- ... to getting the password from the user directly.

Example (Basic guessing)

- Using dictionaries of words.
- Adapt to guesses to password policy, if known.
- ...

Example (Improved guessing)

Take grammar into account, depending on the password type [Bon12; BS12].

Example (Basic guessing)

- Using dictionaries of words.
- Adapt to guesses to password policy, if known.
- ...

Example (Improved guessing)

Take grammar into account, depending on the password type [Bon12; BS12].

Example (Learn from humans)

- Use machine learning [Rip; Cas+17; Wei+09].
- Train algorithm on leaked password databases.
- Generate list of password-looking guesses.

Remark

- This is relevant when the user has chosen a password.
- In the majority of situations it's not.

Example

- There are many devices with default passwords.
- E.g. home routers, ...

Remark

- This is relevant when the user has chosen a password.
- In the majority of situations it's not.

Example

- There are many devices with default passwords.
- E.g. home routers, . . .

Remark

- The problem of default passwords has increased recently.
- Home routers, web cameras are open to attack.

Example (Mirai botnet [Her16])

- Botnet infecting primarily surveillance cameras.
- Attempts default passwords and other vulnerabilities.
- Managed the largest distributed denial-of-service (DDoS) attack hitherto.

Remark

- The problem of default passwords has increased recently.
- Home routers, web cameras are open to attack.

Example (Mirai botnet [Her16])

- Botnet infecting primarily surveillance cameras.
- Attempts default passwords and other vulnerabilities.
- Managed the largest DDoS attack hitherto.

Exercise

- This is a problem when the authentication mechanism faces the Internet.
- E.g. home routers where the admin interface only faces the local network should be fine.
- (The same if we have a white list of addresses allowed access.)
- What do you think?

Example (Autogenerate passwords)

- Generate passwords for users.
- This will likely reduce security by use of post-it notes.
- Not a problem for a home router.

Example (Password ageing)

- Let passwords age and expire.
- Annoying with too short intervals.
- Will reduce security once users introduce systems to remember their last changed password.
- Just an expiration date for the generated one, infinite selected by user.

Example (Rate limiting)

- Remove online guessing by limited login attempts.
- Introduces the possibility of denial of service.

Remark: Offline data

- Consider data which is encrypted with a password.
- You cannot change a password for data that is already stolen.
- You cannot limit the number of attempts either.
- You can just control the guessability of the password.

Exercise

- We now have data to authenticate users.
- How do we store this data?
- What problems do you see?

Example (Password-based authentication)

- Traditionally, there is a password file (or database).
- This contains all users' passwords.
- If someone copies this data, he or she could impersonate any user in the system.

Example (Password-based authentication)

- Traditionally, there is a password file (or database).
- This contains all users' passwords.
- If someone copies this data, he or she could impersonate any user in the system.

Solution (Passwords)

- *We want to compare user-entered and stored password.*
- *We do an irreversible one-way transformation on both.*
- *Then they are still comparable.*
- *The preimage cannot be gained from storage.*

Example

- Cryptographic hash function $h: (\mathbb{Z}_2)^* \rightarrow (\mathbb{Z}_2)^n$.
- On registration, store $h(p)$.
- User authenticates with p' , check if $h(p') \stackrel{?}{=} h(p)$ equals what we stored.

Solution (Passwords)

- *We want to compare user-entered and stored password.*
- *We do an irreversible one-way transformation on both.*
- *Then they are still comparable.*
- *The preimage cannot be gained from storage.*

Example

- Cryptographic hash function $h: (\mathbb{Z}_2)^* \rightarrow (\mathbb{Z}_2)^n$.
- On registration, store $h(p)$.
- User authenticates with p' , check if $h(p') \stackrel{?}{=} h(p)$ equals what we stored.

Remark

- Consider guessing again.
- The used password space is small.
- We only need to evaluate a subset: $h: (\mathbb{Z}_2)^m \rightarrow (\mathbb{Z}_2)^n$.
- With faster computers we can guess a lot.

Solution

- *Choose h to be slow to compute.*
- *E.g. iterate it over itself 10 000 times.*
- *This will slow down guessing attacks.*

Remark

- Consider guessing again.
- The used password space is small.
- We only need to evaluate a subset: $h: (\mathbb{Z}_2)^m \rightarrow (\mathbb{Z}_2)^n$.
- With faster computers we can guess a lot.

Solution

- *Choose h to be slow to compute.*
- *E.g. iterate it over itself 10 000 times.*
- *This will slow down guessing attacks.*

Remark

- The password file structure reveals if two users have the same password.
- Can guess the password for all users at once:
 - 1 Make a guess, compute the hash.
 - 2 Check if it matches *any* user's password.

Solution

- *Add a salt: a small random value (e.g. 128 bits) unique for each user.*
- *Include this value in the computation of the password hash.*
- *Now all hashes will be unique.*

Remark

- The password file structure reveals if two users have the same password.
- Can guess the password for all users at once:
 - 1 Make a guess, compute the hash.
 - 2 Check if it matches *any* user's password.

Solution

- *Add a salt: a small random value (e.g. 128 bits) unique for each user.*
- *Include this value in the computation of the password hash.*
- *Now all hashes will be unique.*

Remark

- The salt is not a secret, it's just unique.
- It can be stored in plain text along with the password hash.

Example

- bcrypt [PM99] implements all this functionality.
- It should also be available in most languages and libraries.

Alternative Approaches

Example (Something you . . .)

- know (passwords)
- have (hardware tokens)
- are (passive biometrics)
- do (active biometrics)

Remark

- Do you *know* a private key or do you *have* one?
- A password you *know*.
- A private key in a hardware token you *have*.
- If the key is stored on your disk?

Example (Something you . . .)

- know (passwords)
- have (hardware tokens)
- are (passive biometrics)
- do (active biometrics)

Remark

- Do you *know* a private key or do you *have* one?
- A password you *know*.
- A private key in a hardware token you *have*.
- If the key is stored on your disk?

Example (Passive biometrics)

- Fingerprint
- Irises
- DNA

Example (Active biometrics)

- Typing speed
- Cursor movement
- Web surfing behaviour
- Pressure points in signature

Example (Passive biometrics)

- Fingerprint
- Irises
- DNA

Example (Active biometrics)

- Typing speed
- Cursor movement
- Web surfing behaviour
- Pressure points in signature

Exercise

- What about something you are (passive biometrics)?
- When is that more than merely a password?

Example (Fingerprints for iPhones)

- The iPhone can trust its built-in fingerprint reader.
- We know that we read an actual finger.
- We know when we read it, so we have freshness.

Example (Fingerprints for web services)

- Fingerprint reader connected to laptop.
- Browser scans and sends fingerprint to server.
- Anyone could send that data, without the reader.
- We lack freshness guarantees.

Example (Fingerprints for iPhones)

- The iPhone can trust its built-in fingerprint reader.
- We know that we read an actual finger.
- We know when we read it, so we have freshness.

Example (Fingerprints for web services)

- Fingerprint reader connected to laptop.
- Browser scans and sends fingerprint to server.
- Anyone could send that data, without the reader.
- We lack freshness guarantees.

Alternative Approaches

Remark

- We need freshness.
- We must store the fingerprint somewhere, to compare.
- Someone can copy that data.
- Without freshness they can use it — as a password.

Remark

- Similar for interactive biometrics.
- Record and replay.

Alternative Approaches

Remark

- We need freshness.
- We must store the fingerprint somewhere, to compare.
- Someone can copy that data.
- Without freshness they can use it — as a password.

Remark

- Similar for interactive biometrics.
- Record and replay.

Exercise

- How can we ensure freshness?
- What data do we need to store for this?
- How can we secure that data?

Solution

- *Freshness is about challenge and response.*
- *Password-based authentication: the same challenge all the time.*
- *Improvement: random challenge, hard-to-guess response.*

Example (Age limits)

- Bob wants to go see a film in cinema.
- Bob looks very young so Alice who works there wants to have proof of his age.
- Show physical ID, reveals name, exact date of birth, . . .

Exercise

- That's a bit overkill, right?
- What does Alice actually need to know?
- In what direction must we move to achieve that?

Example (Age limits)

- Bob wants to go see a film in cinema.
- Bob looks very young so Alice who works there wants to have proof of his age.
- Show physical ID, reveals name, exact date of birth, . . .

Exercise

- That's a bit overkill, right?
- What does Alice actually need to know?
- In what direction must we move to achieve that?

Example (Age limits)

- Bob wants to go see a film in cinema.
- Bob looks very young so Alice who works there wants to have proof of his age.
- Show physical ID, reveals name, exact date of birth, . . .

Exercise

- That's a bit overkill, right?
- What does Alice actually need to know?
- In what direction must we move to achieve that?

What Alice needs?

She must be convinced that Bob is older than 15.

How can she learn that?

- 1 She has known Bob since he was born, so she knows.
- 2 She can ask someone *she trusts* who knows Bob is older than 15.

What Alice needs?

She must be convinced that Bob is older than 15.

How can she learn that?

- 1 She has known Bob since he was born, so she knows.
- 2 She can ask someone *she trusts* who knows Bob is older than 15.

What Alice needs?

She must be convinced that Bob is older than 15.

How can she learn that?

- 1 She has known Bob since he was born, so she knows.
- 2 She can ask someone *she trusts* who knows Bob is older than 15.

But how can she do that?

- 1 The trusted person who knows Bob is with Alice.
- 2 Alice can send a picture to the other person who verifies.
 - This requires an *authenticated* channel.
- 3 The trusted person made a certificate for Bob showing that he's older than 15.
 - Alice must be able to *verify* the certificate.
 - Bob must bring this certificate with himself everywhere.

But how can she do that?

- 1 The trusted person who knows Bob is with Alice.
- 2 Alice can send a picture to the other person who verifies.
 - This requires an *authenticated* channel.
- 3 The trusted person made a certificate for Bob showing that he's older than 15.
 - Alice must be able to *verify* the certificate.
 - Bob must bring this certificate with himself everywhere.

But how can she do that?

- 1 The trusted person who knows Bob is with Alice.
- 2 Alice can send a picture to the other person who verifies.
 - This requires an *authenticated* channel.
- 3 The trusted person made a certificate for Bob showing that he's older than 15.
 - Alice must be able to *verify* the certificate.
 - Bob must bring this certificate with himself everywhere.

Alice interacts with the trusted person

- Gaah, but Bob doesn't want the trusted person (his parents) to know he's at the cinema right now!
- It's a small cinema so they'll know which film he sees if they learn when he's there.

Alice reads and verifies the certificate

- Phew, she accepted the note from his parents.
- But now Alice learned all those embarrassing things in there.
 - And Bob who has a crush on Alice ...

Alice interacts with the trusted person

- Gaah, but Bob doesn't want the trusted person (his parents) to know he's at the cinema right now!
- It's a small cinema so they'll know which film he sees if they learn when he's there.

Alice reads and verifies the certificate

- Phew, she accepted the note from his parents.
- But now Alice learned all those embarrassing things in there.
 - And Bob who has a crush on Alice ...

The idea

- What if Bob could convince Alice
 - that he has a certificate saying he's older than 15,
 - and is signed by someone Alice trusts.
- Wouldn't that be awesome?

Example (Anonymous Credentials¹)

- Makes heavy use of zero-knowledge proofs of knowledge.
- Can prove equalities, inequalities, knowledge, ownership, ...
- Implementations and approaches:

Identity Mixer <https://www.research.ibm.com/labs/zurich/idemix/>

U-Prove <http://research.microsoft.com/en-us/projects/u-prove/>

AnonPass <https://eprint.iacr.org/2013/317>

IRMA <https://www.irmacard.org/irma/>

¹J. Camenisch, A. Lehmann and G. Neven. “Electronic Identities Need Private Credentials”. In: *IEEE Security Privacy* 10.1 (Jan. 2012), pp. 80–83. ISSN: 1540-7993. DOI: 10.1109/MSP.2012.7. URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6142524>

Anonymous Credentials

- [Bon12] Joseph Bonneau. “Guessing human-chosen secrets”. PhD thesis. University of Cambridge, May 2012. URL: http://www.cl.cam.ac.uk/~jcb82/doc/2012-jbonneau-phd_thesis.pdf.
- [BS12] Joseph Bonneau and Ekaterina Shutova. “Linguistic properties of multi-word passwords”. In: *USEC. 2012*. URL: http://www.cl.cam.ac.uk/~jcb82/doc/BS12-USEC-passphrase_linguistics.pdf.

- [Cas+17] Claude Castelluccia, Abdelberi Chaabane, Markus Dürmuth and Daniele Perito. *When Privacy meets Security: Leveraging personal information for password cracking*. 15th Feb. 2017. arXiv: 1304.6584 [cs.CR].
- [CLN12] J. Camenisch, A. Lehmann and G. Neven. “Electronic Identities Need Private Credentials”. In: *IEEE Security Privacy* 10.1 (Jan. 2012), pp. 80–83. ISSN: 1540-7993. DOI: 10.1109/MSP.2012.7. URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6142524>.

- [Her16] Ben Herzberg. *Breaking Down Mirai: An IoT DDoS Botnet Analysis*. Oct. 2016. URL: <https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html> (visited on 18/02/2017).
- [PM99] Niels Provos and David Mazieres. "A Future-Adaptable Password Scheme." In: *USENIX Annual Technical Conference, FREENIX Track*. 1999, pp. 81–91.
- [Rip] John the Ripper community. *John the Ripper bleeding jumbo*. URL: <https://github.com/magnumripper/JohnTheRipper>.

- [Wei+09] Matt Weir, Sudhir Aggarwal, Breno De Medeiros and Bill Glodek. “Password cracking using probabilistic context-free grammars”. In: *Security and Privacy, 2009 30th IEEE Symposium on*. IEEE. 2009, pp. 391–405.