

An Introduction to Privacy Technologies

Prof. George Danezis, UCL

Who am I?



George Danezis

Short bio:

- October 2013 – Today
Professor, University College London
“Security and Privacy Engineering”
- 2007 – 2013
Microsoft Research Cambridge, Researcher & Privacy Champion
- 2000 – 2007
KU Leuven
Cambridge / MIT Institute

Privacy Enhancing Technologies:

- Anonymous communications, traffic analysis, (Tor, Mix, ...)
- Location privacy & economics
- Applied cryptography
- Smart metering Privacy

Email: g.danezis@ucl.ac.uk

Webpage: <http://danez.is>

Resources

Privacy and Data Protection by Design

George Danezis, Josep Domingo-Ferrer, Marit Hansen, Jaap-Henk Hoepman, Daniel Le Métayer, Rodica Tirtea, Stefan Schiffner
ENISA, January 12, 2015

<https://www.enisa.europa.eu/publications/privacy-and-data-protection-by-design>

Privacy as a security property

Security property:

Confidentiality – keeping a person’s secrets secret.

Control – giving control to the individual about the use of their personal information.

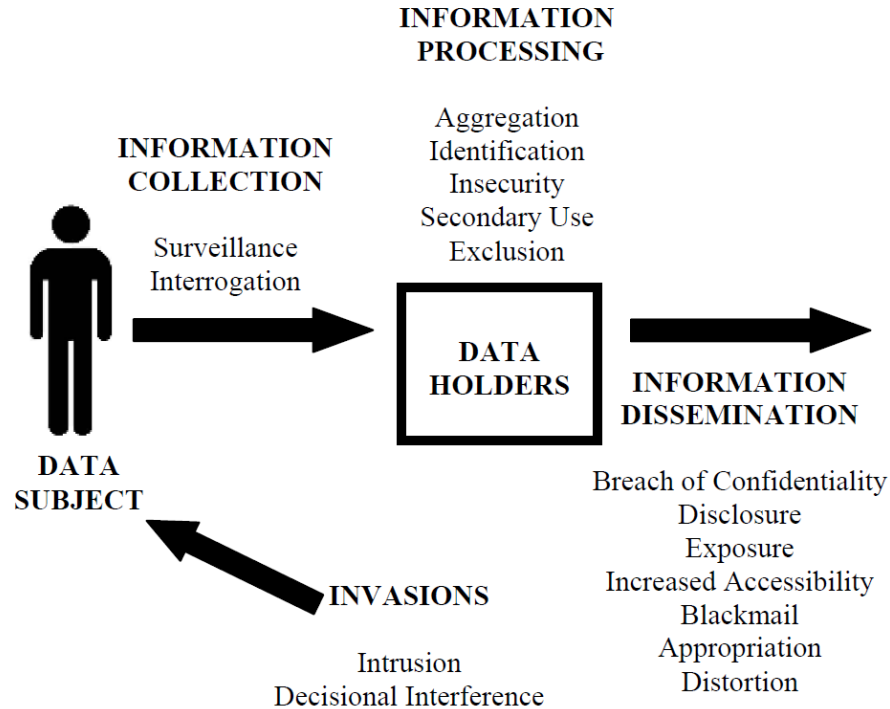
Self-actualization – allowing the individual to use their information environment to further their own aims.

More to privacy:

Sociology, law, psychology, ...

Eg: “The Presentation of Self in Everyday Life” (1959)

Illustrated Taxonomy of Privacy Harms



Taxonomy of privacy harms

A. *Information Collection*

1. Surveillance
2. Interrogation

B. *Information Processing*

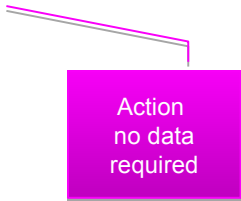
1. Aggregation
2. Identification
3. Insecurity
4. Secondary Use
5. Exclusion

C. *Information Dissemination*

1. Breach of Confidentiality
2. Disclosure
3. Exposure
4. Increased Accessibility
5. Blackmail
6. Appropriation
7. Distortion

D. *Invasion*

1. Intrusion
2. Decisional Interference



Action
no data
required

The Human Right to Privacy

Universal Declaration of Human Rights (1948), Article 12.

No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.

UK Human Rights Act (1998). Article 8.

Right to respect for private and family life

1. Everyone has the right to respect for his private and family life, his home and his correspondence.
2. There shall be no interference by a public authority with the exercise of this right except such as is in accordance with the law and is necessary in a democratic society in the interests of national security, public safety or the economic well-being of the country, for the prevention of disorder or crime, for the protection of health or morals, or for the protection of the rights and freedoms of others.

EU Data Protection Regulations & GDPR

Article 16 of the Treaty on the Functioning of the European Union (Lisbon Treaty, 2009) states that:

1. Everyone has the right to the protection of personal data concerning them.
2. The European Parliament and the Council, acting in accordance with the ordinary legislative procedure, shall lay down the rules relating to the protection of individuals with regard to the processing of personal data by Union institutions, [...]

Personal Data must be processed according to some principles.

How not to engineer for privacy

A step by step guide to bad practices

1. Think of a vague service – no matter how implausible
2. Engineer it to grab and store as such information from users and third parties as possible
3. Hope no one notices or complains
4. When the scandals break out fix your terms of service or do some PR
5. If the scandals persist make your privacy controls more complex
6. When DPAs are after you explain there is no other way
7. Sit on data you have no idea what to do with until your company is sold

Privacy Engineering Principles

Define clearly what you want to do (functional)

Is this by itself privacy invasive?

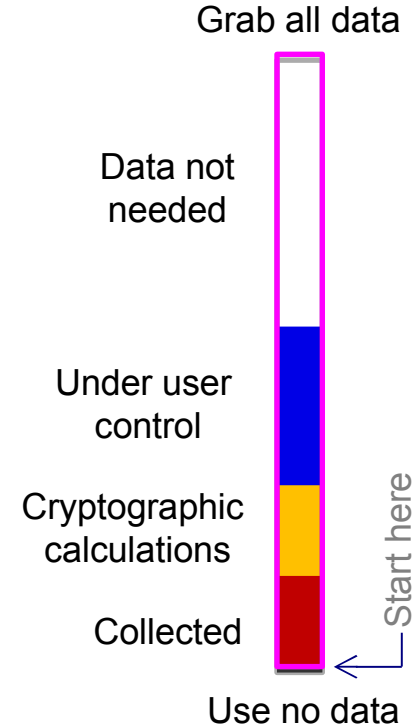
Mechanisms to prevent abuse?

Define the minimum private inputs necessary to achieve the functionality

Build a solution to balance integrity of service and discloses no more information than necessary.

Push processing of private information to user devices

Use advanced cryptography for integrity and privacy



7 principles of Privacy by Design (PbD)

1. Proactive not Reactive; Preventative not Remedial
2. Privacy as the Default Setting
3. Privacy Embedded into Design
4. Full Functionality – Positive-Sum, not Zero-Sum
5. End-to-End Security – Full Lifecycle Protection
6. Visibility and Transparency – Keep it Open
7. Respect for User Privacy – Keep it User-Centric

*“[...] these principles remain **vague** and leave many open questions about their application when engineering systems.” - Gurses et al (2011)*

Privacy Engineering (Gurses et al, 2011)

Process:

Functional Requirements Analysis:

(Vague requirements lead to privacy problems.)

Data Minimization:

(Collecting Identity or PII not always necessary)

Modelling Attackers, Threats and Risks

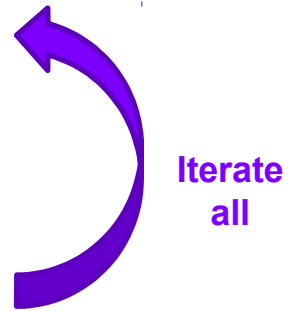
(Which parties have incentives to be hostile to the requirements)

Multilateral Security Requirements Analysis

(Conflicting / contradicting security requirements of all parties)

Implementation and Testing of the Design

← Crucial



“If the functionality was not properly delimited in our case studies, even following our methodology, we would be forced to go for a centralized approach collecting all the data” -- Gurses et al 2009.

PbD and its discontents (I)

“Privacy by design can be reduced to a series of **symbolic activities** to assure **consumers’ confidence**, as well as the free flow of information in the **marketplace**”

“From a security engineering perspective, **control and transparency mechanisms do not provide the means to mitigate the privacy risks** that arise through the collection of data in massive databases.”

Gürses, Seda, Carmela Troncoso, and Claudia Diaz. "Engineering privacy by design." *Computers, Privacy & Data Protection* 14 (2011).

PbD and its discontents (II)

“This becomes especially problematic with respect to large-scale mandatory systems like road tolling systems and smart energy systems, or de facto mandatory systems like telecommunications (e.g., mobile phones).”

Conclusion:

“From a security engineering perspective, the risks inherent to the digital format imply that **data minimization must be the foundational principle** in applying privacy by design to these systems.”

Cryptography & Privacy Enhancing Technologies

A gentle introduction

PETs & their “threat models”

Cryptography is used to build technologies that protect privacy.

Traditional: Confidentiality, control, or even information self-determination.

Privacy a bit different than traditional confidentiality.

What makes Privacy Enhancing Technologies (PETs) different:

- Threat model: weak actors, powerful adversaries.
- Susceptibility to compulsion.
- Cannot assume the existence of Trusted Third Parties (TTP):
- 5Cs: Cost, Collusion, Compulsion, Corruption, Carelessness.

PETs design principles:

- Rely on end-user devices. (Challenge here!)
- Distribute trust across multiple semi-trusted third parties.
- Allow users to chose who they trust for certain operations.
- Use cryptography to ensure confidentiality and correctness.
- Keep only short term secrets, if any.

Perfect Forward Secrecy

Encryption can be used to keep communications secret.

But **what if someone forces you to disclose the key?**

Perfect Forward Secrecy (PFS): gold standard for encrypted communications.

- Start with keys that allow Alice to authenticate Bob, and vice versa.
- Alice and Bob create fresh private/public keys; authenticate and exchange them.
- They establish fresh shared keys, and talk secretly.
- Once done, they delete the shared keys, and fresh private keys.

Result: after a conversation is over, no-one can decrypt what was said.

- Illustrates: using only end devices, no long-term keys.
- Remaining issue: **plausible deniability**.

Available now: Off-the-record (OTR), Signal (Android / iOS), Whatsapp ...

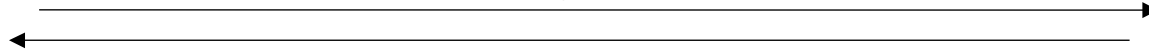
Download “Signal” and use it!

Perfect Forward Secrecy Illustrated



Ver_B
Fresh x
Pub_A = g^x

{ Pub_A }_{sigA}

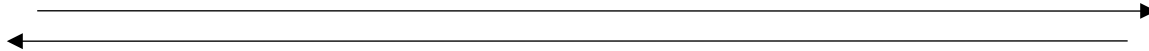


{ Pub_B }_{sigB}



Ver_A
Fresh y
Pub = g^y

K=KDF(g^{xy})



K=KDF(g^{xy})

{ messages }_K



Delete K, x

Delete K, y

Protecting communications meta-data

Who talks with whom, and what you browse is sensitive.

- Alice talks to Bob, Bob is a cancer doctor.
- Alice Browses the NHS website, looking at pages on STDs.

Extensive research shows a lot can be inferred from meta-data:

- Sender, receiver, length & time of communication, pattern.
- Eg. mental condition, personality, language, emotions, political opinion.
- Even if the content is encrypted!

Anonymous communication systems hide such information:

- Best known: **Tor – The Onion Router.**
- How? Use a set of relays:



Illustrates: distribute trust, chose who to trust, crypto ...

Proxies for Anonymous Communications



Alice wants to hide the fact she is sending a message to Bob.

The proxy decrypts the message.

The proxy batches many messages.

The proxy is in the TCB.

Problem:

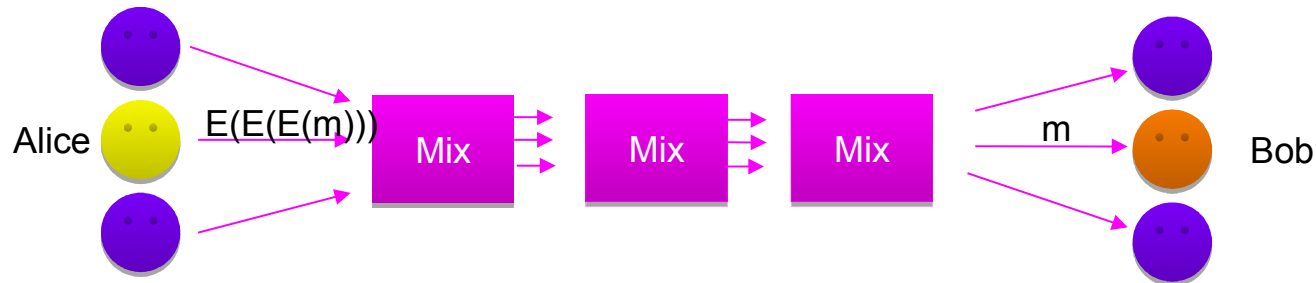
Low throughput.

Corrupt Proxy or Proxy hacked / coerced.

Real case: Penet.fi vs the church of scientology (1996)

Danezis, George, Claudia Diaz, and Paul Syverson. "Systems for anonymous communication." Handbook of Financial Cryptography and Security, Cryptography and Network Security Series (2009): 341-389.

Mix Networks and Onion Routing



Solution: Use multiple cryptographic relays (mix)

Sender encrypts messages using multiple keys, through a set of mixes.

Each mix batches multiple messages.

TCB: Not a single mix, or client. No single place to coerce to trace everyone.

From mix-networks to Onion Routing

OR: sender sends a stream of messages through the sequence of relays.

Problem: timing of traffic leads to correlation (c^2 attack)

Distributed TCB: adversary can compromise some circuits not all.

Private Information Retrieval

Key problem: which database record you access is sensitive!

- Example: which book you are looking at the library?
Which friend you check if they are on-line?
What music you are listening?
Which minister you look up in your online address book?

PETs Solutions:

- **Private information retrieval:** access a public record without leaking which – even to the provider! (Is that even possible?)
- **ORAM:** access your own private encrypted records, without divulging which (cheap) to cloud store.

Techniques: distribute trust, homomorphic encryption, rely on client (e2e).

Private Computations in general

Alice and Bob want to work out who is older, without telling each other their age – can they do that?

- **Amazing result:** any function that could be privately computed by providing the private inputs to a trusted third party, can also be computed privately.
- I.e. Alice and Bob simply exchange cryptographic messages, and end up with the result! Neither of them learns the other's age!
- Also enables **secure outsourcing**.

Two families of techniques:

- **Secure Multiparty Computation:** well established and understood techniques based on secret sharing data.

Commercial support (eg. Cybernetica's Sharemind).

- **Homomorphic Encryption:** allows operations on encrypted data.
Toy Prototypes.

Warning: slow for generic computations.

- Normal CPU 1,000,000,000s (GHz) of operations a second.
- Secure computations 1-10 per second (Hz) in general.

Specific Private Computations

Generic Private Computations slow – but specific ones can be fast.

Smart metering examples: aggregation, fraud detection, billing.

- Private road tolls.
- Private authentication and authorization.
- Simple private statistics.
- Detecting common elements in sets.

Application specific protocols can be practical.

But they need to be evaluated, and the computation needs to be simple.

High-value simple computations are commonplace.

Example deployments: ENCS test-best deployment of privacy-friendly aggregation for smart metering / smart grid roll-outs.

Alfredo Rial, George Danezis, Markulf Kohlweiss: Privacy-preserving smart metering revisited. Int. J. Inf. Sec. 17(1): 1-31 (2018)

Zero-knowledge Proofs & Credential Systems

PETs: 10% confidentiality, 90% making sure no one cheats.

Key: protect users from each other.

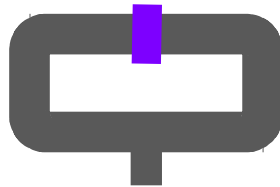
Key: protect users from corrupt elements of the infrastructure.

The challenge: need to prove that something was done correctly, without revealing any private information.

Eg. the electricity bill was computed correctly, but hide how much electricity was consumed at specific times.

Eg. A certifier says I am old enough to have a drink, but I will not tell you my exact age or name.

“Zero-knowledge” proofs – allow you to prove statements about secret values, without revealing them.



How mature are PETs?

Maturity ↑

Not all PETs are equally well understood and mature for use.

PFS: download “Signal” now. Demand it everywhere. 1B users (Whatsapp).

Anonymity: Tor provides a weak form of anonymity, 1M users.

ZKP: Pilots (EU Prime, Primelife, ABC4Trust) & ZCash

Specific Private Computations: pilots (Tor statistics & ENCS smart metering)

PIR / ORAM: we can build it, no large scale deployments.

Generic Private Computations: start-ups & pilots (Cybernetica)

Performance ↑

Performance:

Encryption of communications and storage: super-fast, will not slow down anything you care about.

ZKP: slower, but usually need to prove simple things.

Anonymity / PIR / ORAM: is slower than normal communications.

Private Computations: much slower – 6-9 orders of magnitude.

Privacy Beyond Cryptography

Anonymization, controls on usage, and logging

Other ways to protect privacy

Non-cryptographic technologies are also used to protect privacy.

They have their uses, particularly where a **trusted third party exists.**
Remember the 5Cs: cost, compulsion, collusion, corruption, carelessness.

However some **mechanisms are misunderstood:**

Dataset anonymization.

Query Privacy / Privacy in statistical databases.

Restricting use of collected data.

Logging to detect privacy compromises.

Data Anonymization

“Would it not be nice if: you can take a dataset full of private data, and transform it into one with no private data – while keeping all the value of the data?”

Magical thinking: this cannot happen in general.

The problem of de-anonymization:

- Any aspect of the “anonymized” dataset can be used to link the records to known named records.
- Example of Netflix (anonymous) vs. DBLP (named) de-anonymization.
- In general it is impossible to sanitise only the private information without severely scrubbing all the usefulness out of the dataset.
- Removing PII is not enough!

Data anonymization is a weak privacy mechanism. Only to be used when other (contractual, organizational) protections are also applied.

Arvind Narayanan, Vitaly Shmatikov: Myths and fallacies of "personally identifiable information".
Commun. ACM 53(6): 24–26 (2010) 2006

Query Privacy

“Would it not be nice if I could **send complex queries to a database to extract statistics**, and it returned results that are informative, but **leak very little information** about any individual?”

Possible: state of the art are “differential privacy” mechanisms.

Why is that possible (while anonymization was impossible):

- The final result depends on multiple personal records.
- However it does not depend much on any particular one (sensitivity).
- Therefore adding a little bit of noise to the result, suffices to hide any record contribution.
- In the case of anonymization: need to add a lot of noise to all the entries.

Example: average height in the room via anonymization or query privacy.

Public policy:

- Notice the difference in the shape of the **architecture** to provide robust privacy.

• **Notice that a TTP holds the data**

Cynthia Dwork, Aaron Roth: **The Algorithmic Foundations of Differential Privacy.**

Foundations and Trends in Theoretical Computer Science 9(3-4): 211-407 (2014)

Controls on usage of collected data

“Limiting collection is not practical, so **why not place stricter limits on use instead?**” - Favourite of Craig Mundie (ex-Microsoft)

In practice: use some **access control mechanism** to ensure that once collected the data is only used for some things.

Problems of this approach:

- How does the user, or anyone else, get assurance of robustness?
- Abuses are invisible making this more difficult to police.
- Technically need to keep track of private data and policies – even more complex than ensuring it is not collected.
- Need to ensure consent for use, even more difficult than consent for collection (since user may not even be available – bulk datasets).

Nearly no research on how to robustly achieve this, and prevent abuse.

- Basically: “**trust us** we would never do anything wrong”.
- Transparency efforts at Google Deepmind – internal control.
- **No very clear direction** to design robust technologies.

A cautionary note on more logs

“Well it is simple: you collect all the data, and then you **audit all operations and access to it. Thus if anyone abuses it you can find them and punish them”**

So many problems with this ...

Issues:

- **Authorized accesses are themselves sensitive:**
eg. accesses to medical records. Access to contacts.
- It is not guaranteed that the unauthorized access was not itself the result of a compromised user in the organization.
- **Once personal data leaks it cannot be put back in the bottle.**
Eg. the leakage of private pictures of celebrities.

Public Policy: detecting compromises after the fact is one of the weakest security mechanism, and a weak privacy mechanism. It is not even clear someone can get punished.

Public verifiability and assurance

“How do I know this software I am using provides a gold standard level of privacy protection through PETs?”

Key question!

Answer 1: we leave it up to everyone to examine!

Enormous externality – each user must be an expert and check.

Answer 2: provide clear specifications, require applications providing privacy to provide transparency in their code & operations.

- Concept of “Public verifiability” of both code and operations.
- Gold Standard in the world of PETs (PGP, Tor, Signal, ...)
- Reluctance from industries to adopt for PETs or anything else.
- Serious public policy issue beyond PETs (VW scandal).

At what point does society have a right to know how key machinery works?

- Remember: this was the aim of patents, originally.
- This space will require serious regulation & consumer protection.

In conclusion

Cryptography is everywhere, mostly as a key tool to secure telecommunications and transactions – and privacy.

Cryptographic primitives can be used to build PETs.

- Some of those are very mature (encryption, anonymity systems), and know-how on how to build them more and more commonplace.
- Some are less mature (private computations, PIR), but still practical in special cases.

Some common non-cryptographic privacy protections need careful thought.

- Dataset anonymization and logging are weak at best.
- Query privacy can be robust, given the correct architecture.