


## 8 Privacy and Anonymity

### 8.1 Introduction


**A Definition of Privacy** Recall our discussion on privacy from the first lecture.



**Privacy**

- EU directives (e.g. 95/46/EC) to protect privacy.
- College Bescherming Persoonsgegevens (CBP)
- What is privacy?
  - Users “*must be able to determine for themselves when, how, to what extent and for what purpose information about them is communicated to others*” (Definition PRIME, European project on privacy & ID management.)

2




**Protecting Privacy**

- Hard privacy: data minimization
  - Subject provides as little data as possible
  - Reduce as much as possible the need to *trust* other entities
  - Example: anonymity
- Issues; some information (needs to be) released.
- Soft privacy: *trusted* controller
  - Data subject provides her data
  - Data controller responsible for its protection
  - Example: hospital database medical information
- Issues; external parties, errors, malicious insider

5

Different Privacy Enhancing Technologies (PETs) try to help protect privacy. Here we can distinguish between *hard privacy* and *soft privacy*. In hard privacy the amount of data about subjects is minimized; the user controls the data by not releasing it. In soft privacy users trust certain parties with their data and these data controllers need to protect the data, ensuring it is only used for the right purpose.


### 8.2 Anonymity and Privacy on the internet



**Example: Google**

- “organize the world’s information and make it universally accessible...”
  - Clear risk for privacy; includes personal information
- Multiple services; becoming “omnipresent”
  - Most searches (>90% in NL 2006) but also:
    - Searching books, (satellite) maps, images, usenet, news, scholarly papers, video’s, toolbar, account, email, calendar, photo program, instant messenger
    - Google & Doubleclick adds; used by many websites
    - All linked to IP address user (+os+browser+etc.).

7



**Google’s new privacy policy**

- Combine information different services
  - >60: search, YouTube, Gmail, Blogger, ...
- Could already do for some, now extended

We are confident that our new simple, clear and transparent privacy policy respects all European data protection laws and principle (Quote Google on BBC)

Europe to investigate new Google privacy policy (reuters)

Google privacy changes are in breach of EU law the EU’s justice commissioner has said (BBC)

9

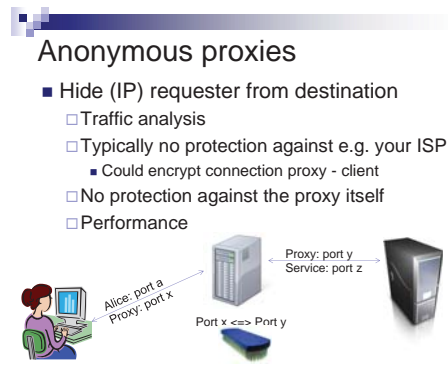
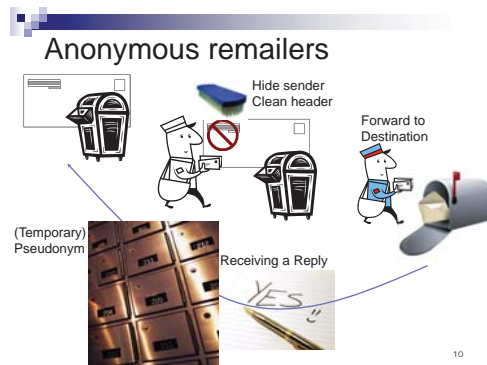
Data is a major asset for most (internet related) companies. Google is a prime example. It’s stated mission of organizing and making universally available the world’s information clearly involves privacy risks. Recent changes to Google’s privacy policy increase its ability to combine data from its many services, allow for extensive profiling of users and consequently creates conflicts with

different privacy legislation.

User data collection, profiling and user tracking (e.g. through cookies) have become so prevalent that law makers have recognized the need to protect users by creating awareness amongst users and offering some measure of control on this data collection (e.g. the new cookie legislation in the Netherlands).

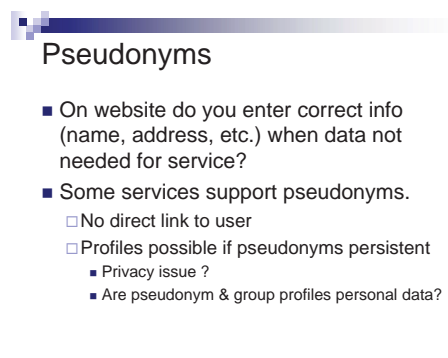
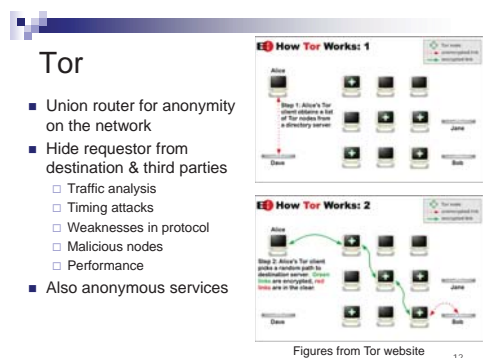
### 8.2.1 Anonymous Surfing

Anonymity is a form of hard privacy; by not giving your identity you can prevent information about you being gathered-or at least prevent the information from being linked to you.



Anonymous remailers allow sending emails without the identity of the sender being revealed to the receiver. You already know a way of doing this your self - recall the lab session in which you spoofed the sender. However, the header information of such a mail would still reveal e.g. the smtp server you used to send (making the mail linkable to the TU/e). Some systems also allow receiving responses by creating a (temporary) mail box the receiver can reply to without being able to link this to the actual sender.

For internet traffic anonymous proxies as 'middle men' in the same way; the server will only see the proxy and not the actual user. Encrypting the data send to the proxy (which includes the server to talk to) would be needed to hide this information from parties that can see your communication; e.g. your internet service provider. Parties that can see the traffic of the proxy may be able to link you to the service your using by looking at the timing of messages (to/from the server/you)



The TOR union router system takes the idea of anonymous proxies a step further; instead of using a single proxy, a chain of proxies is used. All the proxies would have to work together to reveal the link between you and the service. The client selects a sequence of proxies from the list of available proxies. Each proxy in the chain will get the packet content and, encrypted, the address of the next step in the chain and the remainder chain. It decrypts this, sends the remainder of the chain

and the packet content to the next address in the chain. In this way, each proxy in the chain only learns the previous and the next step in the chain.

A main issue with this approach is obviously performance; sending messages through a chain of proxies increases latency (sum of latencies of each connection in the chain) and limits bandwidth (minimum of available bandwidth of each proxy on the chain).


Pseudonyms allow linking different uses of a service (or services) to the same user (the pseudonym) without revealing whom the user (the identity) is. An issue of debate is whether information about the pseudonym (e.g. a user profile) is to be considered personal data. This discussion is complicated by the gray area between pseudonym and identifier. (E.g. is an IP address personal information?) See also the discussion on pseudo-identifiers in the setting of anonymizing databases below (Section 8.3).

### 8.2.2 Zero Knowledge proofs and Direct Anonymous Attestation

Ideally when you authenticate to a service you do not reveal your identity but only the (certified) properties that give you the right to use the service (e.g. your a student at the TU/e). You would want to prove you have a certificate without revealing your identity.

**The magical cave**


- Cave with a fork
- Two passage ways
- Ends of passages not visible from fork



15

**The magical Cave (2)**

- Cave with fork, two passage ways
- Ends of passages not visible from fork
- Ends of passages connected by secret passage way.
- Only findable if you know the secret.




16

**Zero Knowledge proofs** Is this possible at all; can we prove we know a secret without revealing any information about the secret? The answer is Yes. With zero knowledge proofs (ZKPs) we can show in a probabilistic manner that we know the secret. The basic operation of a ZKP can be easily understood through the example of the magical cave [?].

**The magical Cave (3)**

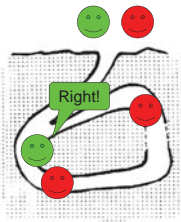
- I know the secret !
- But I won't tell you...
- Can I still convince you I know the secret?



17

**Zero-Knowledge proof**

- Peggy and Victor meet at cave
- Peggy hides in a passage
- Victor goes to the fork
  - calls out either left or right
- Peggy comes out this passage
  - Uses secret passage if needed
- Is Victor convinced ?
  - If repeated many times?



From: Quisquater et al, How to explain Zero-Knowledge Protocols to Your Childre<sub>g</sub>

The setup is as follows: we have a cave which forks into two passages, the ends of which are not visible from the fork. It is a magical cave; the two ends of the passage ways are connected through a secret door which can only be opened with a password. Peggy knows the secret door password and wants to convince Victor that she know it but without telling him the secret.

## Zero Knowledge proof

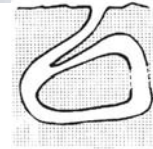
- Peggy convinces Victor she knows secret
- Proof is zero knowledge
  - Consider Victor tapes game
  - Shows tape to you; will you be convinced?
    - Without a proofer who has secret

19

## Example protocol

The Cave:

- Secret  $S$ ,  $p$ ,  $q$  (large primes)
- public  $n = p \cdot q$ ,  $I = S^2 \pmod n$



- P proof knowledge of  $S$  to V
  - P makes random  $R$  sends  $X = R^2 \pmod n$  Peggy hides
  - V makes & sends random bit  $E$  Left/Right
  - P sends  $Y = R \cdot S^E \pmod n$  Peggy comes out
  - V checks  $Y^2 = X \cdot I^E \pmod n$  Victor Sees Peggy

20

This proof is done as follows; prover Peggy and verifier Victor meet at the cave. Peggy enters and hides in one of the passages. Victor goes to the fork and calls out one of the passages (left/right). Peggy comes out of the passage Victor called, using the secret door if needed. After one or even a few successful runs Victor may not be convinced; perhaps Peggy just got lucky and picked the right passage. However, Peggy cannot get lucky all the time; after several repetitions Victor will be convinced.

If Victor tapes this proof and shows it to you later will you be convinced that Peggy knows the secret? Victor could cheat; agree with Eve what he is going to say before hand, or remove failed attempts from the tape. The same tape could be made with Eve who does not know the secret. From this we can conclude that the proof is zero knowledge; you cannot tell the difference between a real and a fake proof - thus could not learn anything from a real proof and the only thing that Victor knows that you do not is that he is not cheating. So Victor also does not learn anything.

## Example protocol analysis

- Completeness
  - With secret  $S$  can always correctly provide  $Y$
- Zero-knowledge; simulation by cheating verifier
  - Simulate run  $(X, E, Y)$ :
 

$X = R^2 \pmod n$   
 $Y = R$  or  $Y = R \cdot S$

    - choose random  $Y, E$
    - if  $E=0$  take:  $X = Y^2$  if  $E=1$  take:  $X = Y^2 / I$
  - Indistinguishable from real runs.
- Soundness ↗ No  $\text{SQRT}(X \cdot S^2)$  and  $\text{SQRT}(X)$  at same time
  - Without  $S$ : Has to choose  $X$  before knowing  $E$ :
    - Choose  $X$  so know  $R = \text{SQRT}(X)$ : No answer if  $E=1$
    - Choose  $X$  so know  $Y = \text{SQRT}(X \cdot S^2)$ : No answer if  $E=0$
  - Thus fails with probability  $1/2$

21

## Use of Zero knowledge proves

- Example protocol show
  - Know secret for given public info
- For applications e.g. DAA
  - Know values with special relation
    - ID along with a CA signature on this ID
  - E.g. know integers  $\alpha, \beta, \gamma$  with properties:
 
$$\text{ZKP}\{(\alpha, \beta, \gamma): y = g^{\alpha} h^{\beta} \wedge y' = g^{\alpha} h^{\gamma} \wedge (u \leq \alpha \leq v)\}$$
    - $\alpha, \beta, \gamma$  secrets,  $y, g, h$ , etc. known parameters
    - $g, h$  generators group  $G$ ,  $g', h'$  for  $G'$

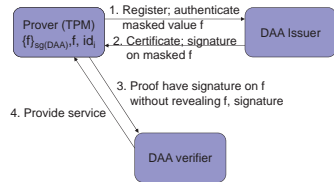
22

This example ZKP system can be used to show possession of a secret  $S$ . The 'hiding in the cave' consists of choosing an  $X$  (which should be constructed by taking the square of some randomly chosen number  $R$ ). Coming out of the correct passage ( $E$  chosen by Victor) consists of providing  $Y = R \cdot S^E$ . Victor can check this value using the public parameters. It is clear that the proof system is complete; Peggy who knows  $S$  can always succeed in proving this by creating the right  $Y$ 's.

About half of the time Victor asks for  $Y = R \cdot S^0 = R$ . Creating this does not require knowing  $S$  so why does Victor not ask for  $Y = R \cdot S$  all of the time? The problem is that if Eve knows that Victor will ask for  $R \cdot S$  she can cheat as follows: instead of taking a random  $R$  and computing  $X$  and  $Y$  from this (which requires the secret) she takes a random  $Y$  and computes  $X = Y^2 / I$  (which does not require the secret). Victor cannot tell how  $X$  is created and the check of  $Y^2 = X \cdot I$  will also succeed. Thus the cases  $Y = R$  are essential; if Eve uses the trick above she will not actually have  $R$ . Thus Eve can either follow the protocol and make  $X$  from  $R$  - in which case she won't be able to answer if Victor chooses 1 or she could create  $X$  from a  $Y$  that she knows but then she won't have  $R$  and will fail to answer if Victor chooses 0.

The correctness argument above also shows why the proof is zero-knowledge; a cheating verifier will be able to create runs which are indistinguishable from the real ones.

### Direct Anonymous Attestation



23

### Direct Anonymous Attestation

- Peggy chooses secret  $f$
- Gets anonymous signature on  $f$ 
  - Does not reveal  $f$  to issuer
  - Recall blind signatures e.g. with RSA
$$E(mr^e) = (mr^e)^d \bmod n = m^{dr} \bmod n = E(m)r$$
- Zero knowledge proof
  - knows an  $f$  together with a signature on  $f$

24

Showing that you know a secret which matches some public information as in this protocol is likely not enough. What you want so show is that you know values with a special relationship (e.g. an identity together with a CA signature on this identity to show that you are a trusted entity without revealing your identity.)

Like with cryptography one would like to be able to build protocols without needing to know the ins and outs of the primitives used. For this we introduce notation e.g. ZKP(secrets, public parameters, properties) to indicate that some system is used to show we know secrets with the given properties. (Like  $[A, B] * pk(A)$  denotes some public key encryption is used without going into the details of the algorithm.)

The Direct Anonymous Attestation (DAA) scheme was adopted by the Trusted Computing Group (TCG) as a privacy preserving method of authenticating a hardware module, the Trusted Platform Module (TPM-in essence a smartcard) embedded in a user's device. The scheme start with the Registration of the Prover. In this step the Prover chooses a secret value  $f$ , and has the issuer sign a masked version of this (recall RSA blind signatures, see Section ??). The Issuer checks Prover's authentication and rights and issues a certificate signing the masked  $f$ .

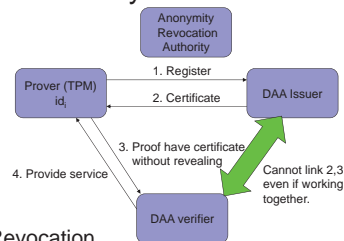
Later when Prover wants to use a service (that is available only to certified users) she authenticates to a Verifier (the service) by proving that she has some  $f$  together with a signature for DAA Issuer on that  $f$ . This proof is zero-knowledge and reveals neither  $f$  nor the signature. (The identity of the Issuer is part of the public knowledge; the Verifier trusts the anonymous user because it trusts the Issuer.) Even working together the Issuer and the Verifier will not be able to determine who the Prover is. DAA does offer the possibility to use pseudonyms, so a Prover can have multiple sessions which can be linked to each other (but not to a specific user).

### Direct Anonymous Attestation

- Rogue member detection / revocation
  - Secret of Peggy =  $f, g$  generator of group
  - Peggy sends  $g^f$
  - Victor
    - Has list revoked  $f'$
    - compares  $g^f$  with  $g^{f'}$  for each on list
    - $g$  not random: not seen to often

25

### Direct Anonymous Attestation



- Revocation
  - of anonymous credentials
  - of anonymity

14

As the TPM is a tamper resistant but not a tamper proof device, the risk that a key may be revealed is considered. If a key/signature pair would be published on the internet then anyone could use it

and, due to the anonymity, the Issuer and Verifier would not be able to tell this known, comprised key it apart from legitimate users. To solve this the Prover also sends  $g^f$  where  $g$  is a generator of the group. Victor can check that  $f$  corresponds to the  $f$  used in the proof but cannot learn  $f$  from this. (Recall that discrete log is considered to be a hard problem.) The Verifier keeps a list of revoked secrets and for each  $f'$  on the list checks whether  $g^{f'}$  is equal to the provided  $g^f$ . Not that same generator  $g$  cannot be used every time as that would make the user traceable (see also section on RFIDs below).

Stronger revocation schemes have been proposed in which certificates for which user certificates for which the key is not known and/or the anonymity of users of a service can be revoked. As the Issues and Verifier cannot do this together (that is the whole point of DAA) these schemes involve a new trusted party with is involved in the registration and gets additional information about the Provers secret. The Prover has to trust that this anonymity revocation authority will only reveal identities/ revoke keys for valid reasons (e.g. abuse of the system).

### 8.2.3 Soft Privacy

Sometimes remaining anonymous is not an option, for example when personal data is needed to be able to deliver a service. In this case we give our data only to trustworthy parties that will protect our privacy. A privacy policy statement describes the intended use of the data; the policy that the company (claims it) will adhere to in using your data. However, typically such privacy policies are not read by the user as often they are complex legal documents and the user has no influence on them; e.g. cannot set preferences.

#### Privacy Policy Statements

- When entering a form on web pages
  - privacy policy: what may be done with data
- Issues
  - To long and complex
  - No guarantees if policy is actually followed
  - No user preferences
    - Accept existing policy / do not use service

27

#### P3P

- Standardized XML based format for privacy policies
  - enables automated tool support
  - e.g. to decide accept cookie
- Issues
  - Policies can be ambiguous
  - No definition how policy should be interpreted
  - Also no enforcement

28

A way to support automated analysis of a privacy policy is the use of P3P; the policies are expressed as XML documents and can thus be machine parsed, e.g. to compare them to a set of preferences the user has indicated. However, a P3P policy is only the specification of the policy, it does not provide a means to implement the policy in the organization.

#### Enterprise Privacy: E-P3P / EPAL

- Mechanisms for enforcement
  - within an enterprise
  - law often requires some for of enforcement
- No External Check
  - For company; ensure employees follow policies
  - User still needs to trust company
- Sticky Policies (policies stay with data)
- Local to company
  - No guarantees outside administrative domain
- Issue: No industry adoption

29

#### Anonymized databases

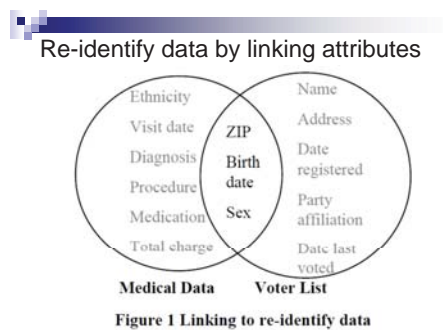
The diagram illustrates the concept of anonymized databases. It features a central purple cylinder labeled 'Medical Records'. To the left, a nurse icon is connected to a folder labeled 'ALICE'. To the right, a doctor icon is connected to another folder. Further right, a red cylinder is labeled 'Attacker Knowledge ("Public" attributes)'. Arrows indicate the flow of information from the medical records to the folders and from the folders to the attacker knowledge.

31

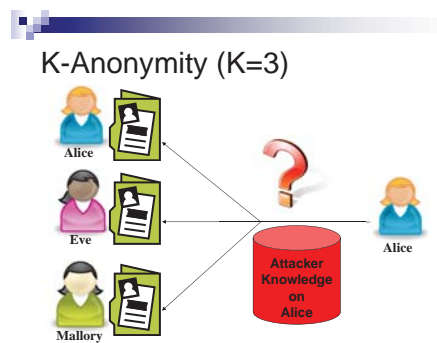
E-P3P and, closely related, EPAL provide mechanisms to enforce P3P policies within an enterprise. The policy is attached to the data so any use of the data can be checked. This allows the enterprise to check/ensure its employees adhere to the policies. The subject of the data still needs to trust the enterprise itself to correctly implement and use this scheme.

### 8.3 Data base anonymization

There are many large databases with personal information that are very useful e.g. for research purposes but the lack of consent by the subjects of the data mean that the data cannot be used as such. One approach is to anonymize the data; remove attributes (e.g. name, social security number etc.) that allow linking the data to an individual. In this way the population can be studied without information being linkable to persons.



k-anonymity: a model for protecting privacy, L. Sweeney in International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 2002 <sup>32</sup>



33

But what information should we remove to correctly anonymize the database? When looking at the data we see datatypes (columns in the data base) include identifiers, (e.g. name, social security number, etc.) that identify the person, non-sensitive quasi-identifiers (e.g. zip code, age, nationality) that give information about the groups that the person belongs to and sensitive personal information (condition, income, etc.) that provide valuable data but should not be linkable to an individual (identifier).

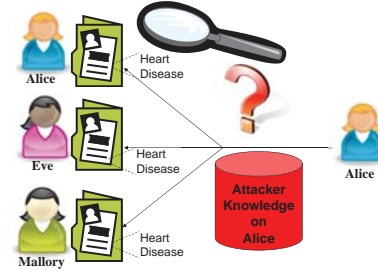
Clearly the identifiers have to be removed. A combination of quasi identifier may also be sufficient to identify a person; e.g. if Alice is the only female born on March 27th in ZIP code 5600 then we know whom this record belongs to and can learn sensitive attributes of Alice. Even if there are only a few entries that match we know one of them relates to Alice which would also be undesirable. To make sure Alice's data is anonymous, she should be indistinguishable (remain 'hidden') amongst a significant number of users. The notion of K-anonymity captures this; every entry should be in a group (equivalence class) of at least  $K$  elements that the attacker cannot tell apart.

Restrict Quasi-ids to achieve

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	130**	< 30	*	Heart Disease
2	130**	< 30	*	Heart Disease
3	130**	< 30	*	Viral Infection
4	130**	< 30	*	Viral Infection
5	1485*	≥ 40	*	Cancer
6	1485*	≥ 40	*	Heart Disease
7	1485*	≥ 40	*	Viral Infection
8	1485*	≥ 40	*	Viral Infection
9	130**	3+	*	Cancer
10	130**	3+	*	Cancer
11	130**	3+	*	Cancer
12	130**	3+	*	Cancer

Fig. 2. 4-anonymous inpatient microdata. I-Diversity: Privacy Beyond k-Anonymity by A. Machanavajhala et al. in ACM Transactions on Knowledge Discovery from Data 2007 34

Attribute Disclosure

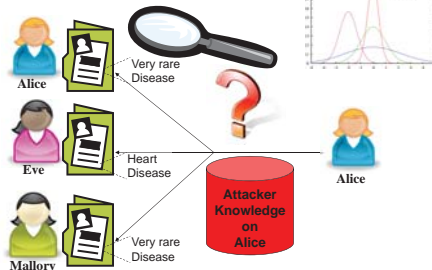


35

K-anonymity can be achieved by removing quasi-identifiers or giving less details on their values (e.g. an age range instead of the exact age). But even if we do this, Alice's anonymity is not guaranteed; if all members in her equivalence class (i.e. the records that could be Alice's records from the perspective of the attacker) have the same value for the sensitive attribute then the attacker still learns the value of the sensitive attribute for Alice. The L-diversity property addresses this; the equivalence class should have  $L$  different values for the sensitive attribute considered. (One can consider simple counting; at least  $L$  different values occur or one can also take into account how often these values appear requiring the entropy to be at least  $L$ .)

Still we may be revealing probabilistic information about Alice's sensitive attribute. If the distribution of values in Alice's equivalence class deviates significantly from the distribution of values for the whole population. The T-closeness requires these distributions to be similar.

Probabilistic disclosure



36

K-anonymity, L-diversity, T-closeness

- Equivalence class in released DB
  - Records that an attacker cannot tell apart
  - Same value for attributes known to attacker
- K-anonymity; in each equivalence class
  - at least K members
- L-diversity; in each equivalence class
  - at least L possible/likely values for attribute
- T-closeness; in each equivalence class
  - Distribution attributes similar to global distribution

37

## 8.4 RFIDs and user tracing

Radio Frequency Identification (RFID) systems are a wireless technology for automatic identification consisting of a set of tags, readers and a backend. The tags are typically very simple devices consisting of a tiny chip and an antenna offering very limited resources. The readers are connected with the backend, which stores all the relevant information about the tags. The tags interact with the readers through identification protocols that aim to provide the identity of the tag to the backend system in a secure manner.



### RFID system

- Wireless technology for automatic identification
  - a set of tags
  - a set of readers
  - a backend
- Identification protocols
  - Specify interaction tags & readers
  - goal: *securely* get identity of the tag to backend
- Readers connected with the backend
  - Backend stores valuable information about tags

39

### Application

- Supply chain automation
- Warehouses (real-time inventory)
- Medical applications
- (People) tracking
  - security tracking for entrance management
- Timing
  - (sports event timing to track athletes)



40

This technology has been employed for an increasing number of applications, ranging from barcode replacement to electronic passport. The basic idea is always to wirelessly identify an object to a centralized system, be it an access control system or a shelf management system. Although RFID systems are only a small subset of the existing identification systems, they have unique advantages that made them attractive for several uses. For example RFID tags are already used in libraries to speed up book loans, in buses to automate check-in and check-out for subscribers, or at the entrance of buildings to legitimize people to enter. On the other hand, the wireless nature of RFID makes access to tags extremely easy, introducing several issues. For example, an attacker can eavesdrop or even start communications to tags to analyze their outputs and obtain sensitive information. Unlike in many other kinds of identification systems, such information also includes the location of tags, as they may travel with their owner. Therefore, an attack which does not identify a tag but does distinguish it from other tags is already an issue, because it allows the attacker to track its location.

### Privacy problems

Why?

- ease of access (wireless nature)
- constrained resources
- extensive use

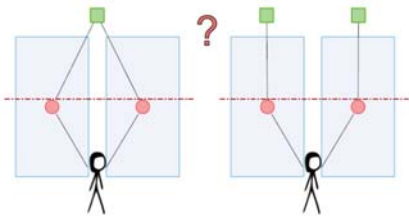
→ leakage of information about the owner's behaviour

Desired Properties?

- untraceability
  - adversary cannot link two sessions to same tag
- forward privacy
  - adversary cannot link past sessions of stolen tag
- backward privacy, etc.

41

### Untraceability game



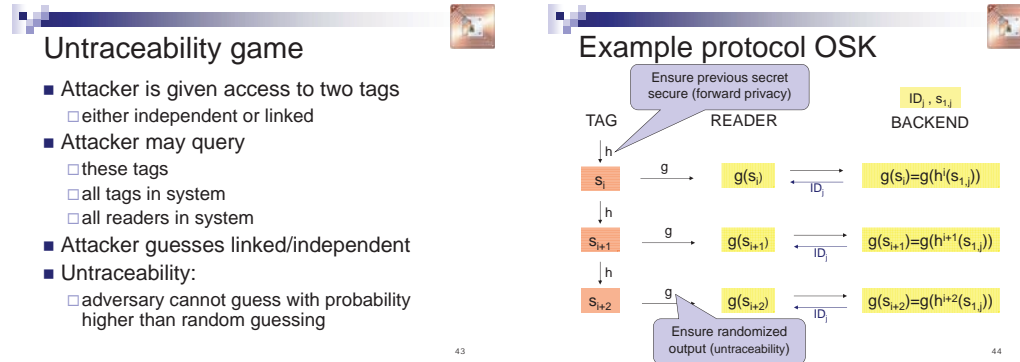
The diagram illustrates the untraceability game. It shows two sessions, each represented by a blue box containing a stick figure (the adversary) and a green square (the tag). In the first session, the tag is connected to two red circles (readers). In the second session, the tag is connected to two different red circles. A question mark is placed between the two sessions, indicating the adversary's goal is to determine if the tag is the same in both sessions.

42

The problem of information leakage leads to security goals of untraceability, forward and backward privacy, properties that hold when the attacker is not able to gain any information about tags. Untraceability means that an attacker should not be able to trace the movement of a tag, i.e. observing past events should not allow an attacker to distinguish between tags. The stronger goal of forward privacy in turn becomes important when the attacker can obtain the tag in question, e.g. by stealing it or even simply buying the item it is attached to. As tags are simple devices, the attacker can break the tag to obtain any information stored in it. Still, this should not enable the attacker to trace the tag in retrospect, i.e. to learn its past locations. Also in the backward privacy analysis, the attacker is given the ability of tampering with a tag. The goal of this privacy property is to prevent the attacker from tracking a tag once she obtained its secret information.

We can precisely express properties such as untraceability using security games, just like we did for the security properties of encryption (recall CPA, CCA-2, etc.). The security game for untraceability gives the attacker two interfaces (e.g. RFID readers) and one of two possible situations that the attacker should not be able to tell apart; either both interfaces actually talk to the same tag or

they each talk to their own tag. If the attacker, who may also query all tags and readers in the system cannot guess which situation occurs (with a probability higher than random guessing) then untraceability is satisfied.



A protocol that achieves untraceability and forward privacy is the OSK protocol. It uses two different hash functions. Hash function  $g$  is applied to the current secret and the result is sent to the reader. The hash function ensures the attacker is not able to obtain the secret from this message. The hash function  $h$  is then used to update the state of the tag, replacing its secret  $s$  by  $h(s)$ . If we would keep the same state  $s$  then the next message we send would again by  $g(s)$  which allows an attacker to trace the tag. For untraceability any change of the state to a new value, e.g. to  $s + 1$ , would be sufficient as the attacker cannot tell that  $g(s + 1)$  and  $g(s)$  are hashes of related values. The stronger property of forward privacy does rely on  $h$  being a one-way function; if an attacker steals the tag and extracts its current secret  $s_i = h(s_{i-1})$  she will still not know previous secret  $s_{i-1}$  and cannot find which of the sessions she has previously observed belong to this tag.

The back-end has the initial secrets  $s_{1,j}$  for each tag  $j$  (along with the identity  $ID_j$  linked to this tag). It can identify the  $i$ -th session of tag  $j$  by computing  $g(h^i(s_{1,j}))$  and comparing this to the received message. (For efficiency it will likely store  $s_{i,j} = h^i(s_{1,j})$  rather than (only)  $s_{1,j}$ .)

## 8.5 Conclusions and where to go from here

In this chapter we have seen several privacy properties and PETs. Privacy, Pseudonymity and Anonymity are sometimes confused. As we have seen in this chapter anonymity can be a PET but is not the only way of preserving privacy. Also the use of pseudonyms / data not including explicit identifiers does not imply that there will be no privacy issues.

Privacy plays a role in several master courses within the Kerckhoffs program. The Seminar Information Security Technology address RFIDs while the Privacy Seminar (Nijmegen) focusses specifically on privacy. The Security and Privacy in Mobile Systems (Twente) and Secure data management (Twente) address PETs while Law in Cyberspace (Nijmegen) looks at privacy from a legal perspective.

### 8.5.1 Literature

Suggested reading (check the course page [2] for the most up to date list of suggested reading materials):

- Security Engineering Introduction [3, Ch 20]. Section 20.4 on Privacy Protection.
- Privacy Policies [6]. Compares P3P, E-P3P, and audit logic.
- t-Closeness: Privacy Beyond k-Anonymity and l-Diversity [9].

- Zero knowledge proofs; introduction [?], Actual system(s) [11].
- Handbook of applied cryptography [10, Ch 10]. Section 10.4 on zero knowledge proofs.

## 8.6 Exercises

1. Consider again the online music store of the previous chapters. Review your requirements analysis, adding privacy consideration; threats and countermeasures where appropriate. You should have a reasonable security requirements description now. Review your complete requirements, design options and choices making sure they make sense as a combination. Work out a secure basic design of the system.
2. Consider the following table. To anonymize this data we need to remove the identifier (name) but also prevent re-identification by other known attributes, the quasi-identifiers.

Identifier	Quasi-identifiers				Sensitive Attribute
Name	Zip Code	Age	Gender	Nationality	Expertise
Alice	5600	30-45	female	Dutch	security
Bob	5600	30-45	male	Dutch	security
Colossus	1000	60+	female	English	computation
Dave	1000	30-45	male	American	astronomy
Eve	5600	30-45	female	Dutch	cyber crime
Fran	1000	60+	female	American	computation
Gill	1000	30-45	female	English	astronomy
Hall	1000	30-45	male	American	computation
Isaac	5600	30-45	male	Greek	astronomy
Julia	1000	30-45	female	Italian	security
Mallory	5600	30-45	female	Dutch	cyber crime

- (a) Which quasi-identifiers need to be hidden to achieve 2-anonymity?
  - (b) Which quasi-identifiers need to be hidden to achieve 2-diversity? (Counting based rather than entropy based.)
  - (c) Even if we hide the quasi-identifiers given in a and b, the distribution in the groups does not really match overall distribution (t-closeness is not satisfied for a reasonable value  $t$ ); in which classes are there expertises that are much more likely than that same expertise in the general population?
3. Consider the following approach to protecting a database containing a membership list indexed by lastname, containing fullname, address, email, etc. in the data field.
    - Every index field is replaced by the hash of the lastname.
    - Every data field is encrypted with a symmetric algorithm using the lastname as the key.
    - (a) Can you effectively find the information about Jan Pietersen?
    - (b) Against which risk does this defense try to protect?