An overview of privacy concepts in computer science

Prashant Agrawal*

Department of Computer Science and Engineering, IIT Delhi

*Joint work with Anubhutie Singh, Malavika Raghavan, Prof. Subodh Sharma and Prof. Subhashis Banerjee

Agenda

To give an overview of privacy techniques in computer science and evaluate how well they align with the legal principles of privacy

- Only a broad-strokes picture (privacy research almost 4 decades old, dozens of journals and conferences, thousands of PhDs)
- Details and references in Sec. 3 of *"An operational architecture for privacy-by-design in public service applications"* by P. Agrawal, A. Singh, M. Raghavan, S. Sharma and S. Banerjee. (Link: https://arxiv.org/pdf/2006.04654.pdf)

Informational privacy, data security and data protection

- ***Informational privacy:*** The broad concept of individuals' right to be left alone (Puttaswamy I)
- ***Data security:*** The technical safeguards and operations kept in place by entities to protect the data that is collected and stored by them.
- ***Data protection:*** A legal framework for achieving informational privacy, e.g., by preventing unlawful collection and processing by entities.

Legal principles of privacy (OECD)

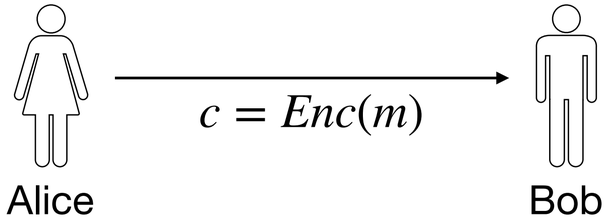
- ***Collection Limitation:*** Collection should occur with the knowledge and consent of the individual
- ***Data Quality:*** Only relevant and necessary information is collected
- ***Purpose Specification:*** Intended purpose of data collection must be specified
- ***Use Limitation:*** Data shouldn’t be used for purposes other than those specified at collection
- ***Individual Participation***
- ***Security Safeguards***
- ***Openness***
- ***Accountability***

Privacy risks

1. Leakage of sensitive data in transit
2. Unauthorised access of information
3. Linking of information shared across multiple databases
4. Re-identification of individuals from anonymised data
5. Post-access manual purpose violation (insider attacks)
   - e.g., data selling, illegal surveillance, misuse of data
6. Post-access automatic purpose violation
   - e.g., illegal profiling or targeting using AI

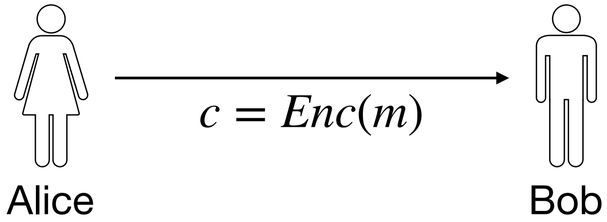
Encryption

- Protects data in storage & transit
  - given c, hard to find m (e.g., hardness of factoring in RSA - Rivest et al. ’78)

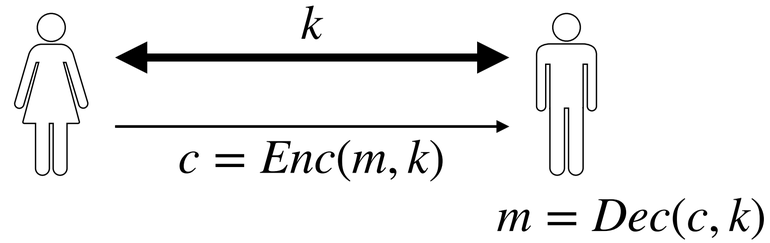


Encryption

- Protects data in storage & transit
  - given c, hard to find m (e.g., hardness of factoring in RSA - Rivest et al. ’78)

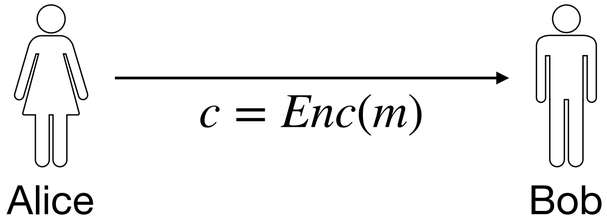


- Symmetric encryption (aka shared-key encryption)

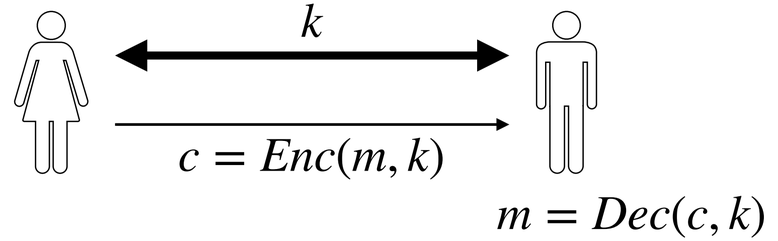


Encryption

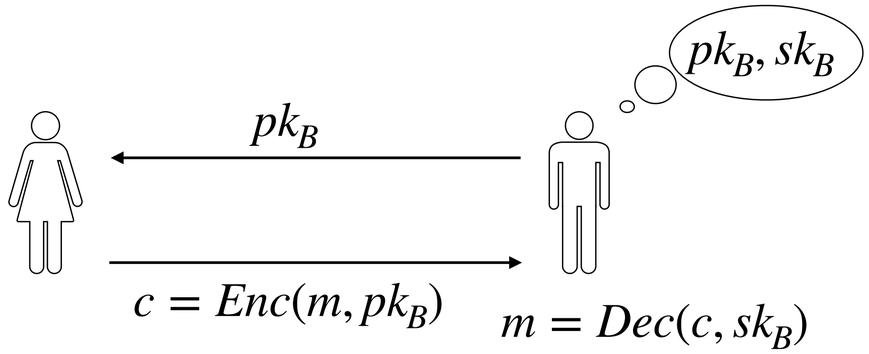
- Protects data in storage & transit
  - given c, hard to find m (e.g., hardness of factoring in RSA - Rivest et al. ’78)



- Symmetric encryption (aka shared-key encryption)

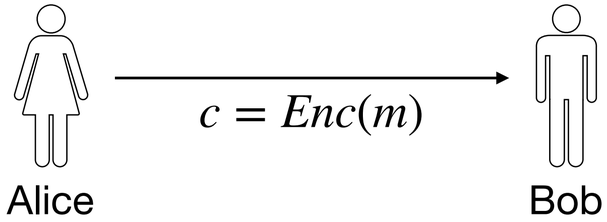


- Asymmetric encryption (aka public-key encryption)

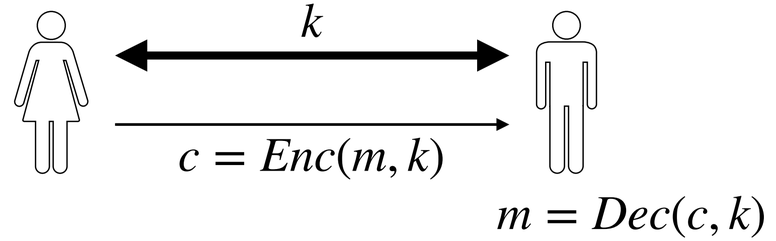


Encryption

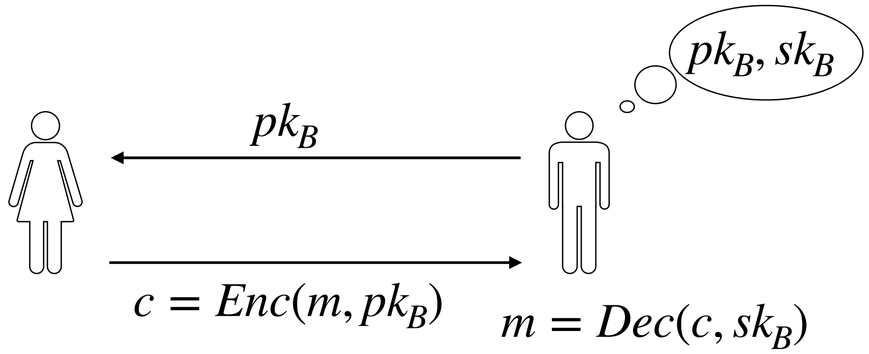
- Protects data in storage & transit
  - given c, hard to find m (e.g., hardness of factoring in RSA - Rivest et al. ’78)



- Symmetric encryption (aka shared-key encryption)



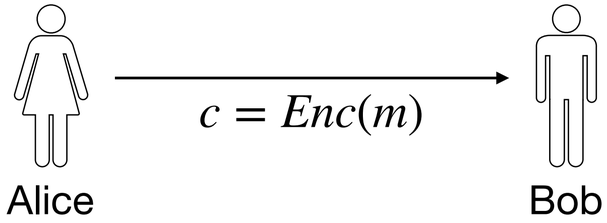
- Asymmetric encryption (aka public-key encryption)
  - Symmetric encryption above is typically achieved by exchanging k using asymmetric encryption (e.g., Diffie-Hellman key exchange - Diffie & Hellman ’76)



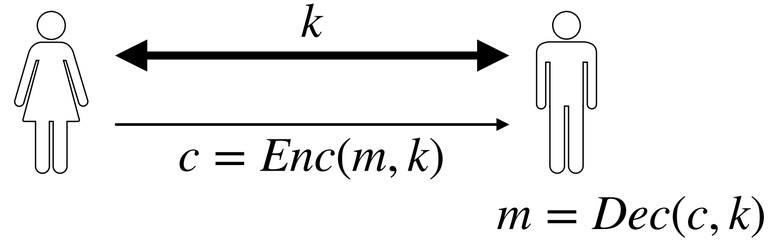
Encryption

Computation still requires decrypting to plaintext

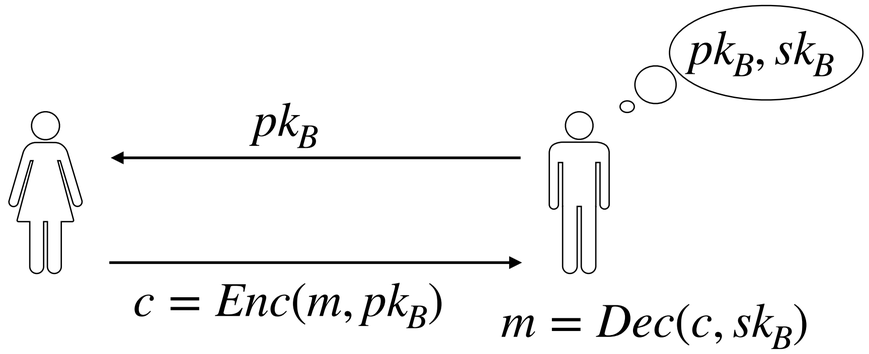
- Protects data in storage & transit
  - given c, hard to find m (e.g., hardness of factoring in RSA - Rivest et al. ’78)



- Symmetric encryption (aka shared-key encryption)

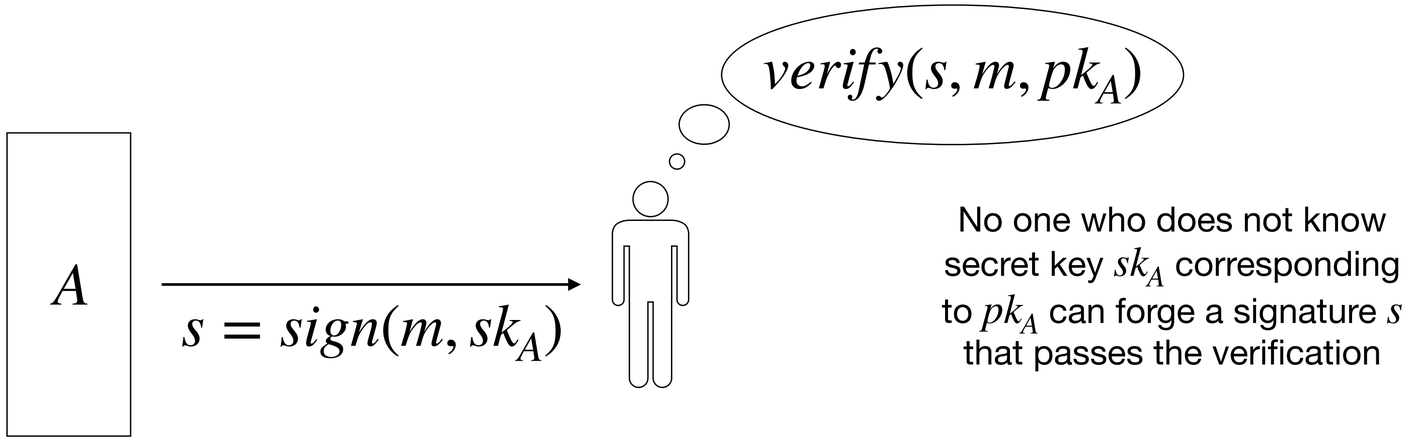


- Asymmetric encryption (aka public-key encryption)
  - Symmetric encryption above is typically achieved by exchanging k using asymmetric encryption (e.g., Diffie-Hellman key exchange - Diffie & Hellman ’76)



Digital signatures

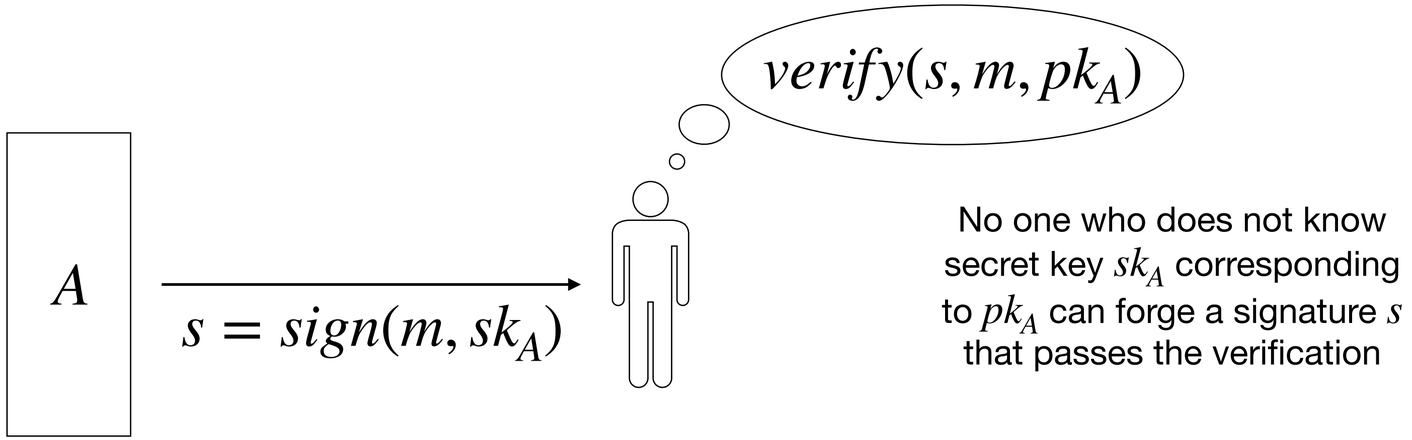
- Signatures make messages authentic



- Signatures are also non-repudiable (A cannot later deny to a third party that it did not sign m; s is an evidence that it did)

Digital signatures

- Signatures make messages authentic

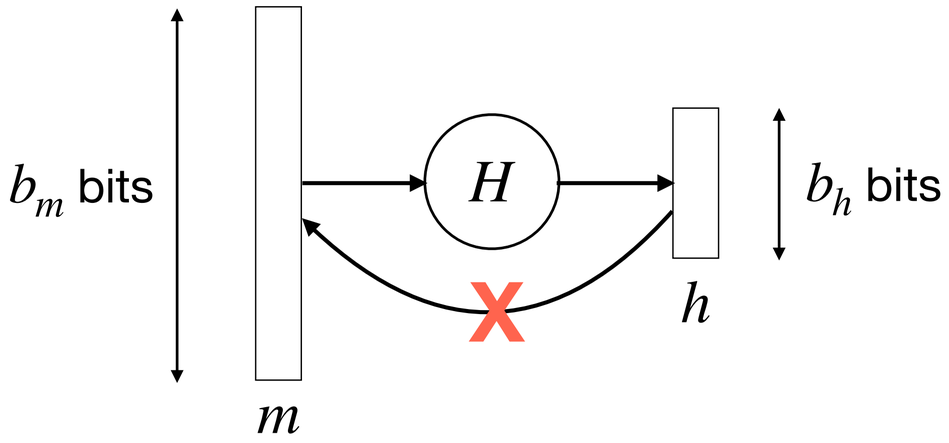


- Signatures are also non-repudiable (A cannot later deny to a third party that it did not sign m; s is an evidence that it did)

Security of encryption and digital signatures crucially depends on who has access to the secret keys!

One-way hash functions

- Given message m, computing hash $h = H(m)$ is easy, but given h, it's hard to find which m might have produced it (h appears to have been *randomly generated*).



- *Collision resistance:* It's even hard to find another m' such that $h = H(m')$
- E.g., md5, sha256, etc.

Data minimisation (DM) (aka “minimum disclosure”)

- Share only the minimum amount of data required for the purpose
  - e.g., to prove that “I am over 18 years” without disclosing my exact DOB or unique ID
  - e.g., sharing only anonymised information for statistical purposes

Data minimisation (DM) (aka “minimum disclosure”)

- Share only the minimum amount of data required for the purpose
  - e.g., to prove that “I am over 18 years” without disclosing my exact DOB or unique ID
  - e.g., sharing only anonymised information for statistical purposes

- Many techniques and concepts:
  1. Zero-knowledge proofs
  2. Anonymity and unlinkability: virtual identities, anonymous credentials, etc.
  3. Database anonymisation

DM1: Zero-knowledge proofs

- Techniques to prove a statement *without revealing anything other than the statement itself* (Goldwasser et al. ’89)
  - e.g., to prove that “I know the secret key corresponding to a given public key”, without revealing the secret key

Sudoku in Zero Knowledge

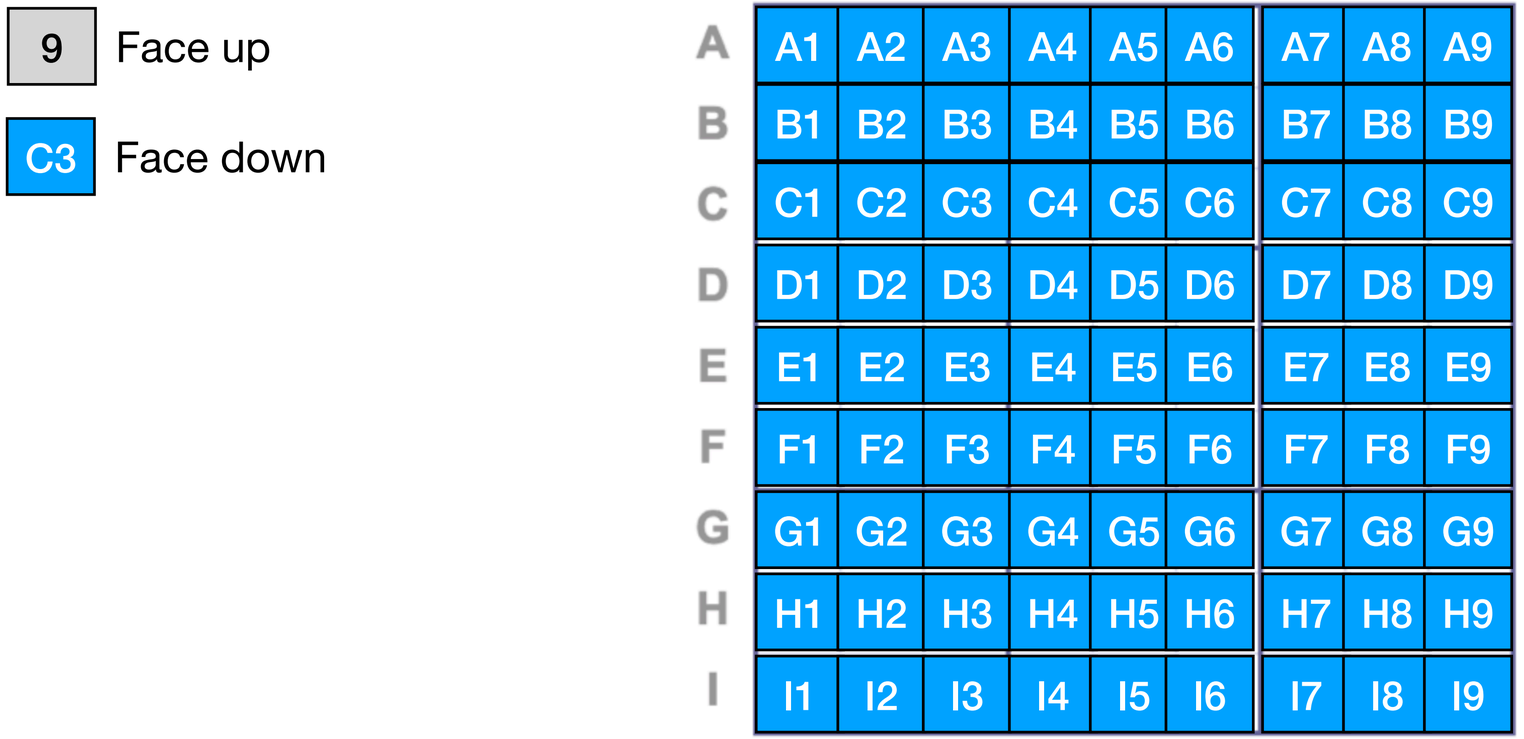
Goal: *Prover wants to prove to the verifier that it knows the solution* to the following Sudoku puzzle, without revealing the solution to the verifier.*

	1	2	3	4	5	6	7	8	9
A							6	8	
B					7	3			9
C	3		9					4	5
D	4	9							
E	8		3		5		9		2
F								3	6
G	9	6					3		8
H	7			6	8				
I		2	8						

* For a correct solution to a Sudoku problem, each row, column and box must contain all the numbers from 1-9

Sudoku in Zero Knowledge

- Step 1 (**Commitment**): Prover writes the solution for each cell in one of the 81 cards placed face down over the Sudoku grid. Since the cards are numbered, the prover is *committed* to the location of each card.

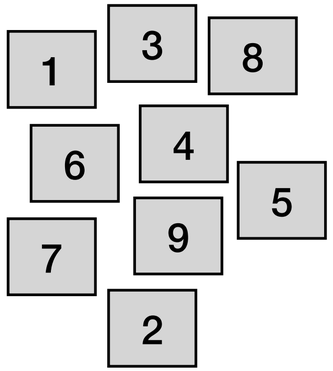


Note: Commitment is both **binding** and **hiding**!

Sudoku in Zero Knowledge

- Step 2 (**Challenge/Response**): Verifier challenges to open a random column/row/box. Prover shuffles the cards in the requested column/row/box and reveals face up. Verifier checks that all 1-9 are present and all other face-down cards are intact. Repeat k times.

Shuffled cards of the first column revealed face up!



	1	2	3	4	5	6	7	8	9
A		A2	A3	A4	A5	A6	A7	A8	A9
B		B2	B3	B4	B5	B6	B7	B8	B9
C	3	C2	C3	C4	C5	C6	C7	C8	C9
D	4	D2	D3	D4	D5	D6	D7	D8	D9
E	8	E2	E3	E4	E5	E6	E7	E8	E9
F		F2	F3	F4	F5	F6	F7	F8	F9
G	9	G2	G3	G4	G5	G6	G7	G8	G9
H	7	H2	H3	H4	H5	H6	H7	H8	H9
I		I2	I3	I4	I5	I6	I7	I8	I9

Original problem

	1	2	3	4	5	6	7	8	9
A							6	8	
B					7	3			9
C	3		9					4	5
D	4	9							
E	8		3		5		9		2
F								3	6
G	9	6					3		8
H	7			6	8				
I		2	8						

Sudoku in Zero Knowledge

- Step 3 (**Final Reveal**): Prover reveals all non-empty cells in the original Sudoku problem to show that he has actually solved the given problem, and not some other problem.

Verifier is convinced that prover knows the solution but does not learn *anything* about the solution.

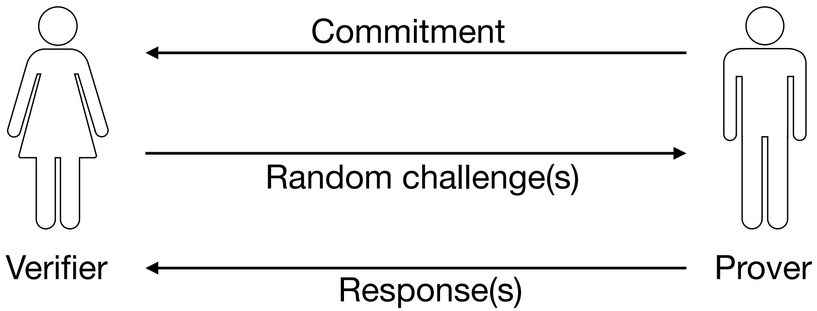
	1	2	3	4	5	6	7	8	9
A	A1	A2	A3	A4	A5	A6	6	8	A9
B	B1	B2	B3	B4	7	3	B7	B8	9
C	3	C2	9	C4	C5	C6	C7	4	5
D	4	9	D3	D4	D5	D6	D7	D8	D9
E	8	E2	3	E4	5	E6	9	E8	2
F	F1	F2	F3	F4	F5	F6	F7	3	6
G	9	6	G3	G4	G5	G6	3	G8	8
H	7	H2	H3	6	8	H6	H7	H8	H9
I	I1	2	8	I4	I5	I6	I7	I8	I9

Original problem

	1	2	3	4	5	6	7	8	9
A							6	8	
B					7	3			9
C	3		9					4	5
D	4	9							
E	8		3		5		9		2
F								3	6
G	9	6					3		8
H	7			6	8				
I		2	8						

Zero-knowledge proofs: Summary

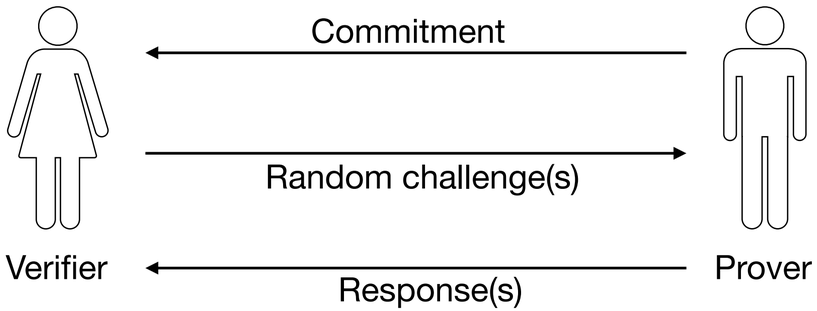
- Typical structure



- Commitment with binding+hiding properties is a cryptographic construct too (Pedersen ’91)
- Interactive ZKPs can be made non-interactive using one-way hash functions instead of random challenges (Fiat & Shamir ’86)

Zero-knowledge proofs: Summary

- Typical structure

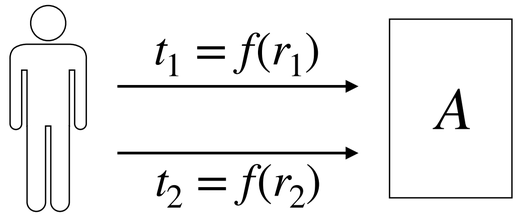


- Commitment with binding+hiding properties is a cryptographic construct too (Pedersen ’91)
- Interactive ZKPs can be made non-interactive using one-way hash functions instead of random challenges (Fiat & Shamir ’86)
- **All practical statements* can be proved in zero knowledge with overwhelming probability** (Goldreich et al. ’91)

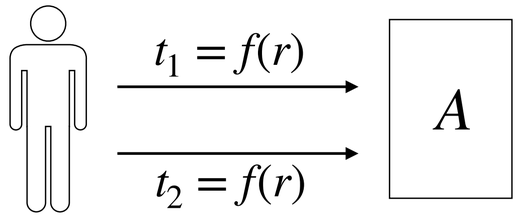
* All NP statements, i.e., which can be verified in polynomial time

DM2: Anonymity and unlinkability

- *Anonymity:* The state of not being identifiable in a set of individuals
- *Unlinkable anonymity:* Transactions do not reveal individuals' true identities and even multiple transactions by the same individual are unlinkable

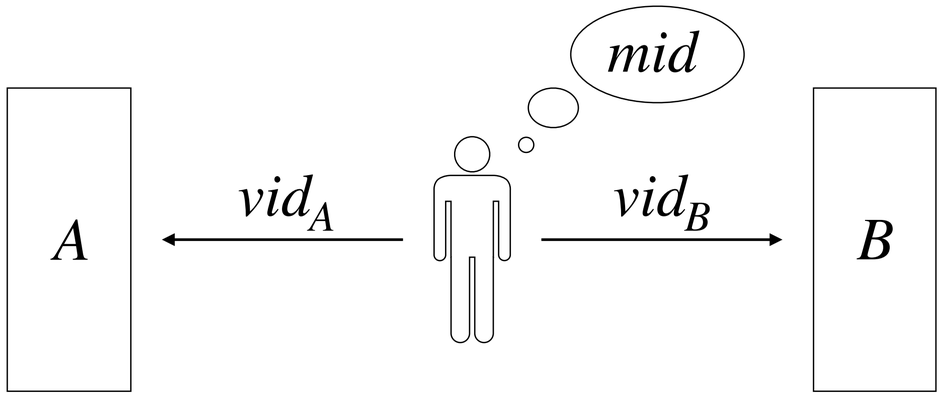


- *Linkable anonymity:* Transactions do not reveal individuals' true identities but multiple transactions by the same individual are linkable



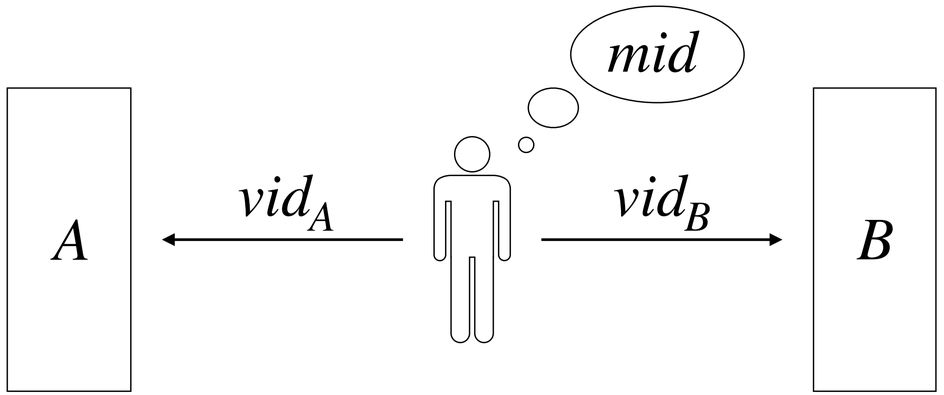
Virtual identities

- Individuals own a master identity and generate random looking, completely unlinkable virtual identities for different organisations (Chaum ’85)



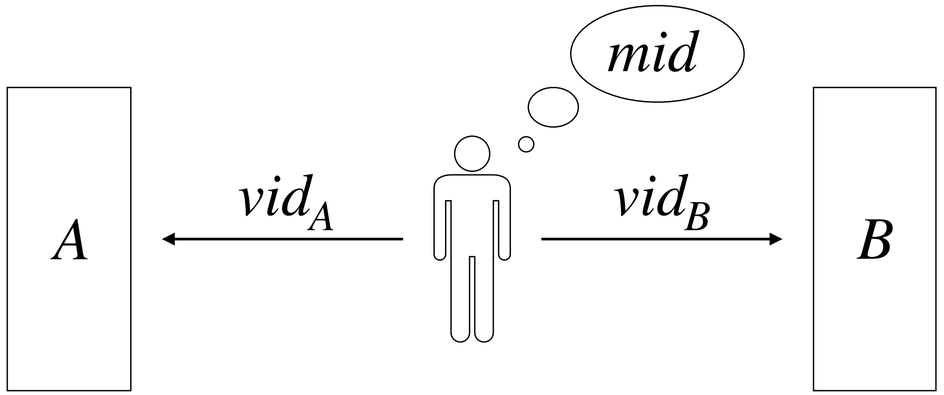
Virtual identities

- Individuals own a master identity and generate random looking, completely unlinkable virtual identities for different organisations (Chaum ’85)



- Unlinkable anonymity for inter-organisation transactions / linkable or unlinkable anonymity for intra-organisation transactions
- Purpose-limited linkability by a trusted authority (for genuine reasons, accountability)

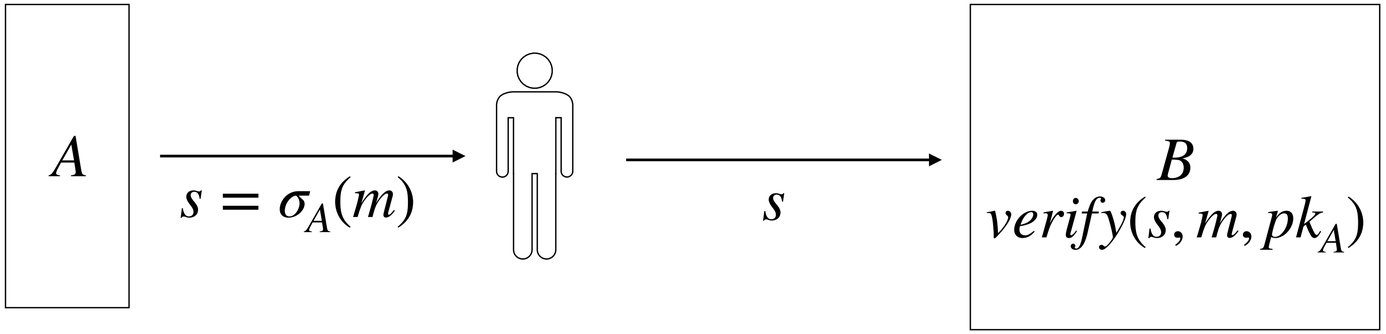
Anonymous credentials



- What if the individual wants to show some credentials obtained from A to B, without allowing A or B to link vid_A and vid_B (e.g., A=college, B=employer)?

Anonymous credentials

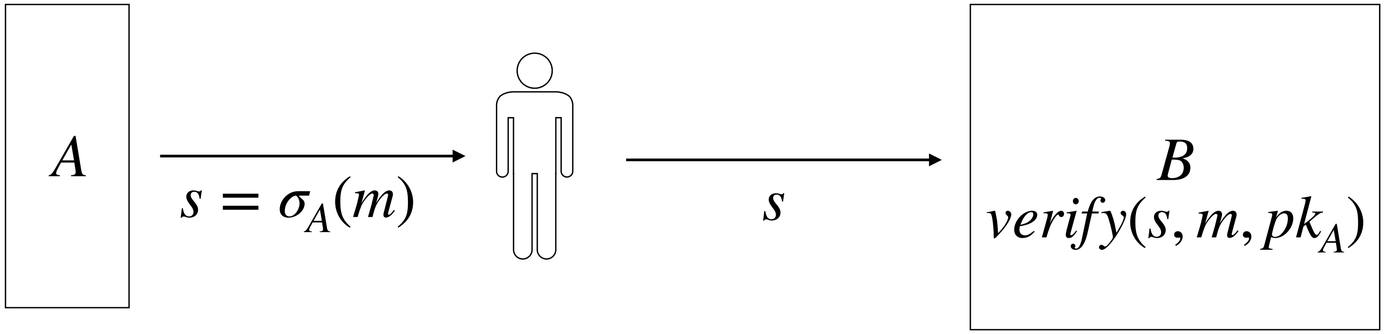
- Regular credentials: Based on *digital signatures* that are unforgeable



No one except A can forge a signature that passes the verification

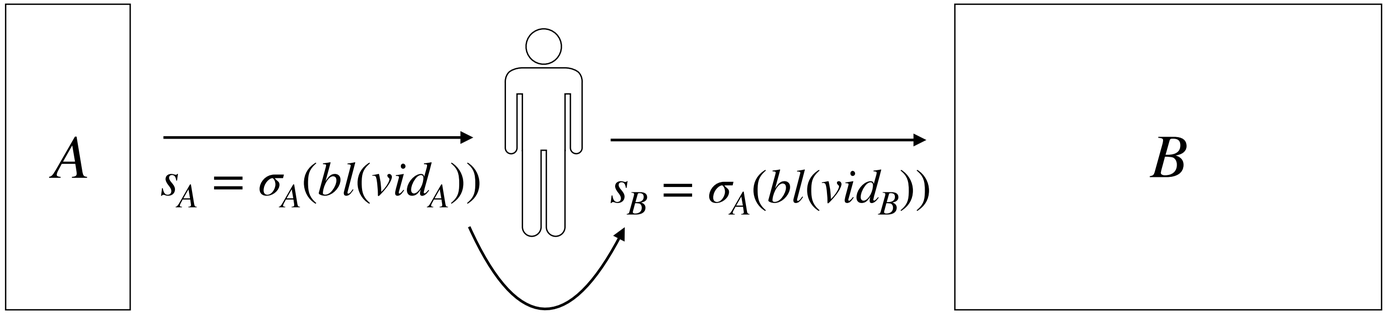
Anonymous credentials

- Regular credentials: Based on *digital signatures* that are unforgeable



No one except A can forge a signature that passes the verification

- Anonymous credentials: Often based on *blind signatures* that are transformable (Chaum ’85)



No one can present a signature on a vid unless they obtained a signature on another vid they own

e.g. $s_A = \sigma_A(''vid_A$ has degree X$'')$ ⟶ $s_B = \sigma_A(''vid_B$ has degree X$'')$

DM3: Database anonymisation

- Data minimisation technique to allow analytics on DBs while preserving anonymity
- Hide personally identifiable information by adding noise, suppressing info or coarsening data

Name	Age	Sex	Height	Weight	Location	HIV
Alice	15	F	5.5	84	Rohila Apartments, Pune	Yes
Bob	28	M	5.1	58	Bldg X, DLF Phase 3, GG	No
Charles	34	M	5.9	65	Sameer Bungalow, Delhi	No
David	43	M	6.1	76	55, Sunset Blvd, Mumbai	No

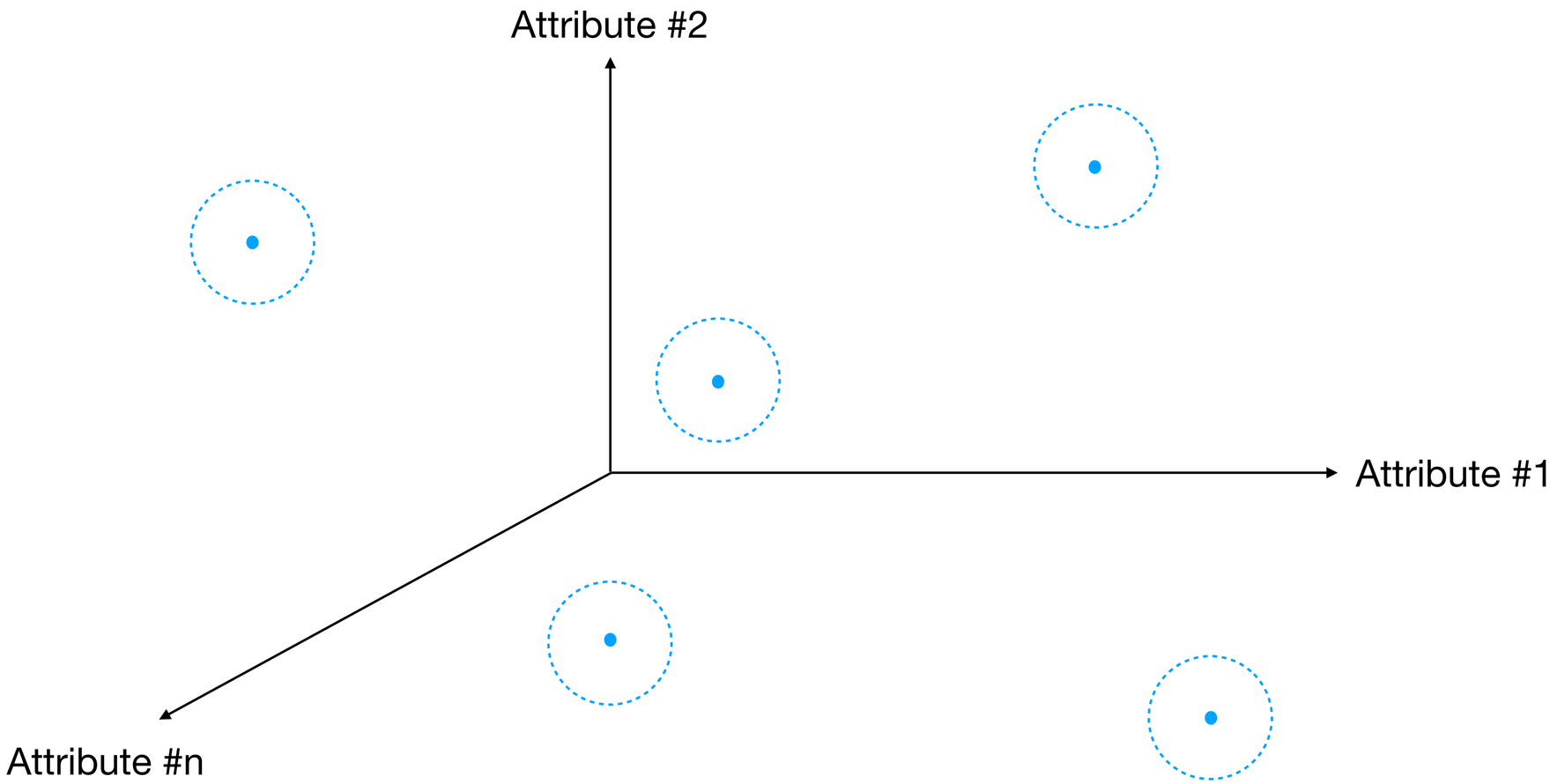
Medical Database

Name	Age	Sex	Height	Weight	Location	HIV Status
*	15-20	F	5.5-6.0	80-85	Pune	Yes
*	25-30	M	5.0-5.5	55-60	Gurgaon	No
*	30-35	M	5.5-6.0	60-65	Delhi	No
*	40-45	M	6.0-6.5	75-80	Mumbai	No

Anonymised Medical Database

- Many notions: k-anonymity, l-diversity, etc.

Anonymisation is a myth



Individuals map to points in a sparse high-dimensional space where they are uniquely identifiable even after adding a lot of noise.

Anonymisation is a myth

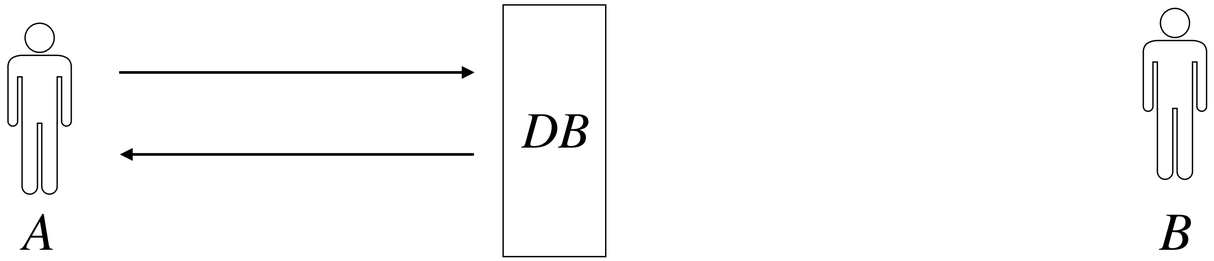
- De-anonymisation attacks on anonymised social network data, location data, writing style, source code, browser history, etc., exist. (Narayanan et al. ’19)

- *Theoretical bottlenecks:* Given a database with n rows, if the adversary is allowed to obtain answers to $O(n)$ subset-sum queries, it can ***reconstruct the entire database*** (unless you add an unacceptable amount of noise) (Dinur & Nissim ’03)

- *Rough intuition:* Solve a bunch of linear equations to derive individual values from aggregate answers

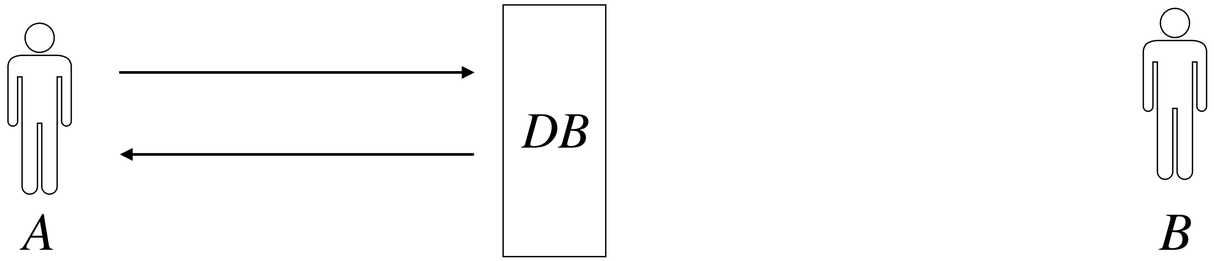
Impossibility of absolute privacy

Absolute privacy goal (aka inferential privacy): A should not obtain any information about an individual that B cannot obtain without access to DB



Impossibility of absolute privacy

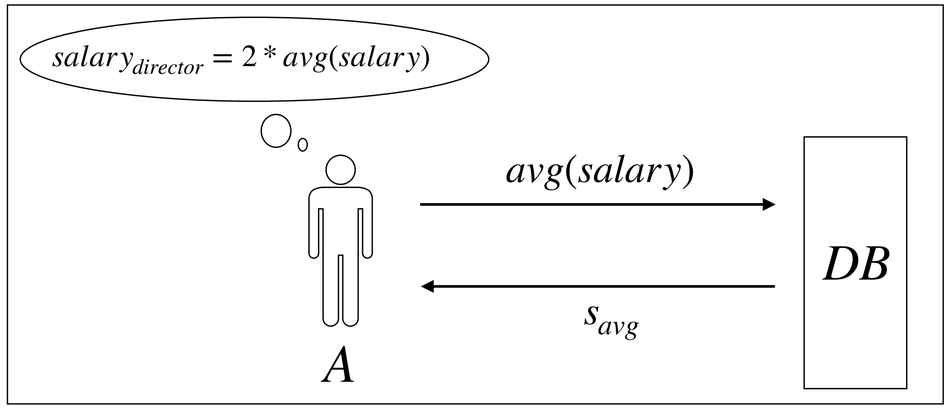
Absolute privacy goal (aka inferential privacy): A should not obtain any information about an individual that B cannot obtain without access to DB

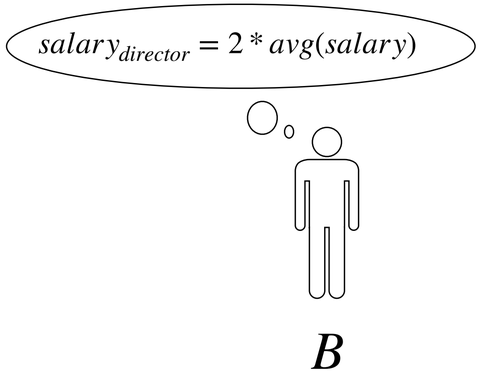


If the adversary has arbitrary side-information, above absolute privacy goal is impossible to achieve. (Dwork ’05)

Impossibility of absolute privacy

Absolute privacy goal (aka inferential privacy): A should not obtain any information about an individual that B cannot obtain without access to DB

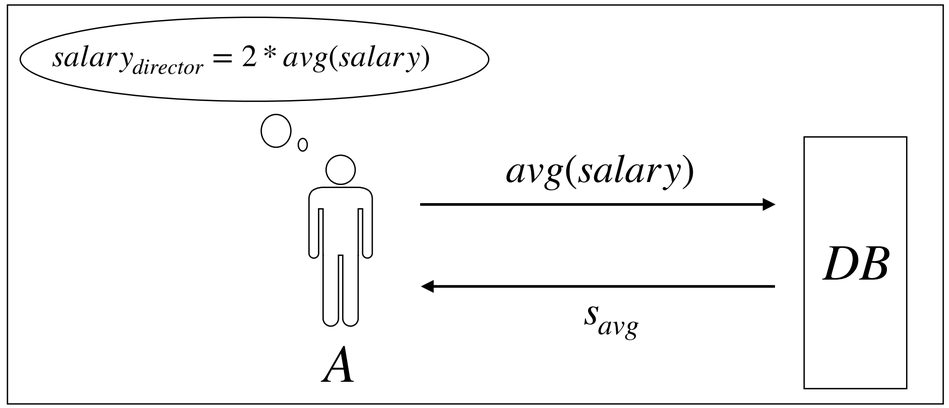


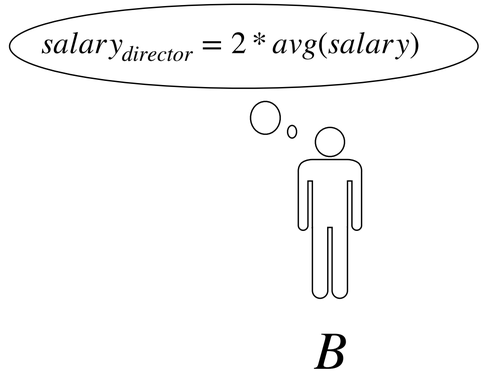


If the adversary has arbitrary side-information, above absolute privacy goal is impossible to achieve. (Dwork ’05)

Impossibility of absolute privacy

Absolute privacy goal (aka inferential privacy): A should not obtain any information about an individual that B cannot obtain without access to DB



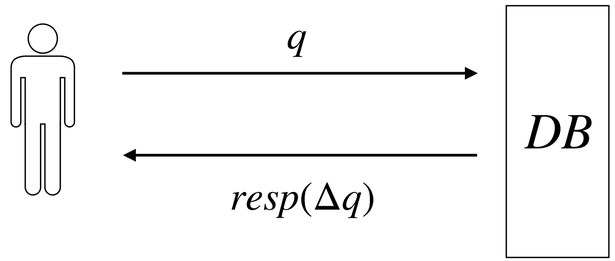


If the adversary has arbitrary side-information, above absolute privacy goal is impossible to achieve. (Dwork ’05)

***Observe:** Privacy of director’s salary is compromised even if the director is not in the DB*

Changing the goalpost: differential privacy

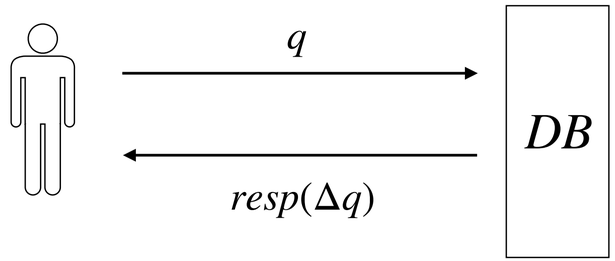
Differential privacy goal: Database should answer such that the *additional* privacy risk any individual incurs by participating in the database is minimal.



Query q's sensitivity Δq measures how much the answer for q can change w.r.t. changes in only one row

Changing the goalpost: differential privacy

Differential privacy goal: Database should answer such that the *additional* privacy risk any individual incurs by participating in the database is minimal.



Query q's sensitivity Δq measures how much the answer for q can change w.r.t. changes in only one row

Interactively calibrate noise as per query sensitivity to maintain user's differential privacy

Limitations of differential privacy

- As query sensitivity increases, noise increases
- DP mechanisms for different types of queries need to be specially designed
- Cannot easily answer many many queries (although some sophisticated noise addition techniques allow answering many queries of a special type)
- How to prevent community-level profiling (cf. Cambridge Analytica)?
- Not suitable for non-statistical uses

Necessary conditions for privacy

- Impossibility of absolute privacy suggests that *all illegal data accesses and processing must be prevented in the first place*.

- Data controllers must declare purpose upfront and mechanisms should exist to only allow computations that fulfil the stated purpose.

- Legitimate purpose depends on dynamically changing consent, approvals, authentication, etc.

- Preliminary work on purpose-based privacy policies exists but it mostly assumes a poor proxy for purpose (e.g., role of the data requester)

- Also, data minimisation should be followed as a further defence and whenever data exits the regulatory boundary

Secure remote execution

Goal: *Data should be secured such that a remote party can only execute a given program on it, and not use it in any other way.*

- **Software solutions (fully secure, but slow):**
  - Homomorphic encryption (computing in ciphertext space without decrypting)
  - Secure multiparty computation and garbled circuits

- **Hardware solutions (depend on trusted hardware, but fast):**
  - Intel SGX (many of today's laptops, desktops, etc.)
  - ARM Trustzone (Android devices)

Homomorphic encryption (computing without decrypting)

Additive homomorphic encryption

$$Enc(a + b) = Enc(a) \oplus Enc(b)$$

Multiplicative homomorphic encryption

$$Enc(a \times b) = Enc(a) \otimes Enc(b)$$

Fact: *All computations can be expressed as a circuit of $+$ and $\times$.*

Homomorphic encryption (computing without decrypting)

Additive homomorphic encryption

$$Enc(a + b) = Enc(a) \oplus Enc(b)$$

Multiplicative homomorphic encryption

$$Enc(a \times b) = Enc(a) \otimes Enc(b)$$

} ***Fully homomorphic encryption (FHE)***

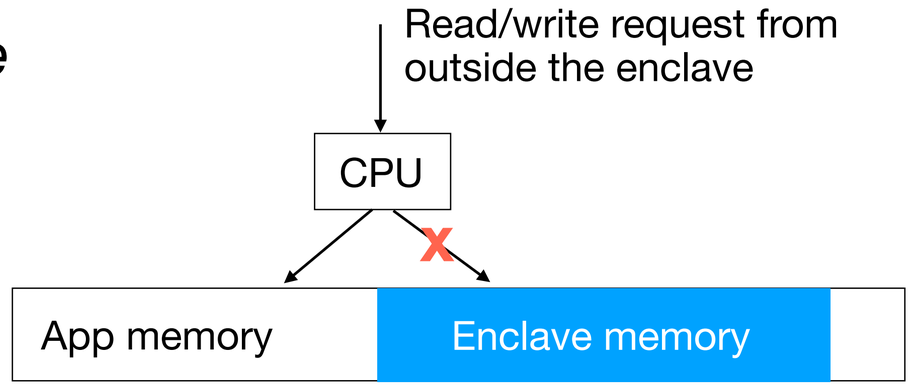
Fact: *All computations can be expressed as a circuit of $+$ and $\times$.*

Creating an encryption scheme Enc that satisfies both the above equations is tough. Such schemes exist but are extremely slow. Also, conversion to circuit model isn't practical.

Intel SGX

Confidentiality: *No one can peek into any state of the program running within the enclave.*

Integrity: *No one can modify execution of the program running within the enclave, except through explicit entry points.*

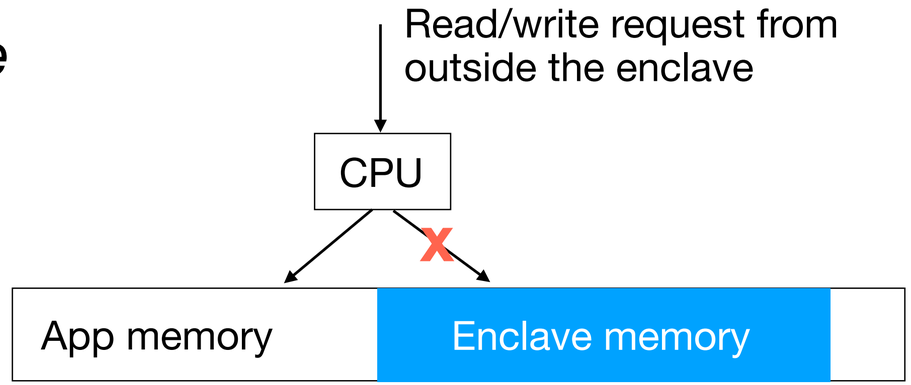


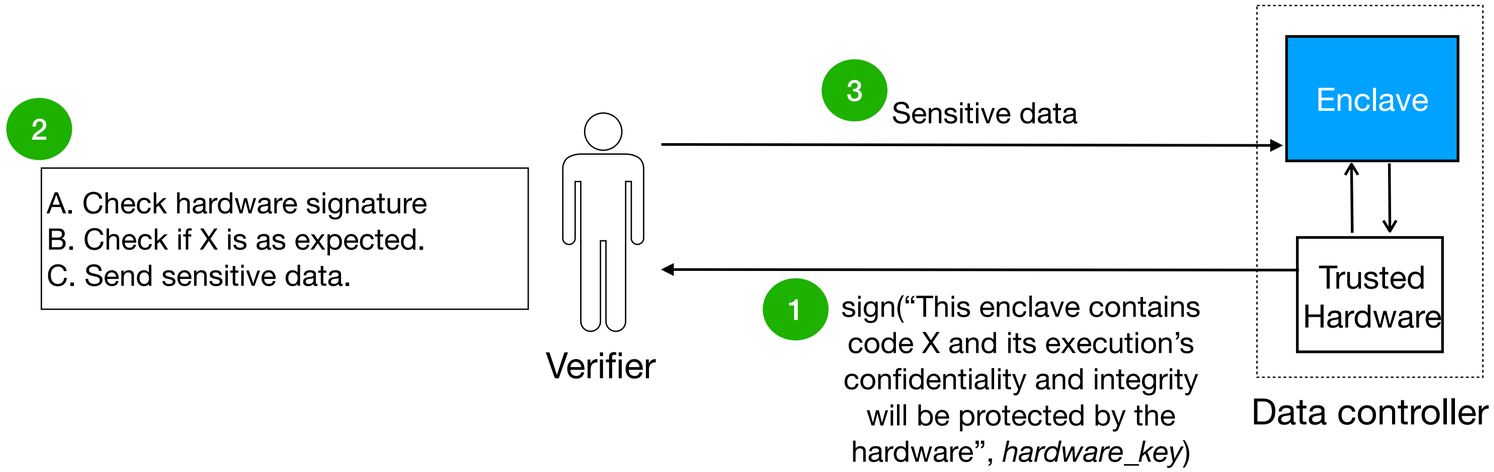
Intel SGX

Confidentiality: *No one can peek into any state of the program running within the enclave.*

Integrity: *No one can modify execution of the program running within the enclave, except through explicit entry points.*

Remote attestation: *Sensitive data will only be processed as per code X.*





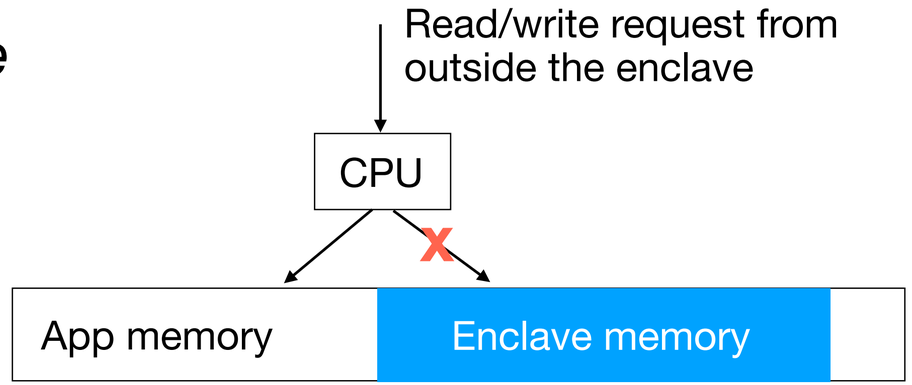
Intel SGX

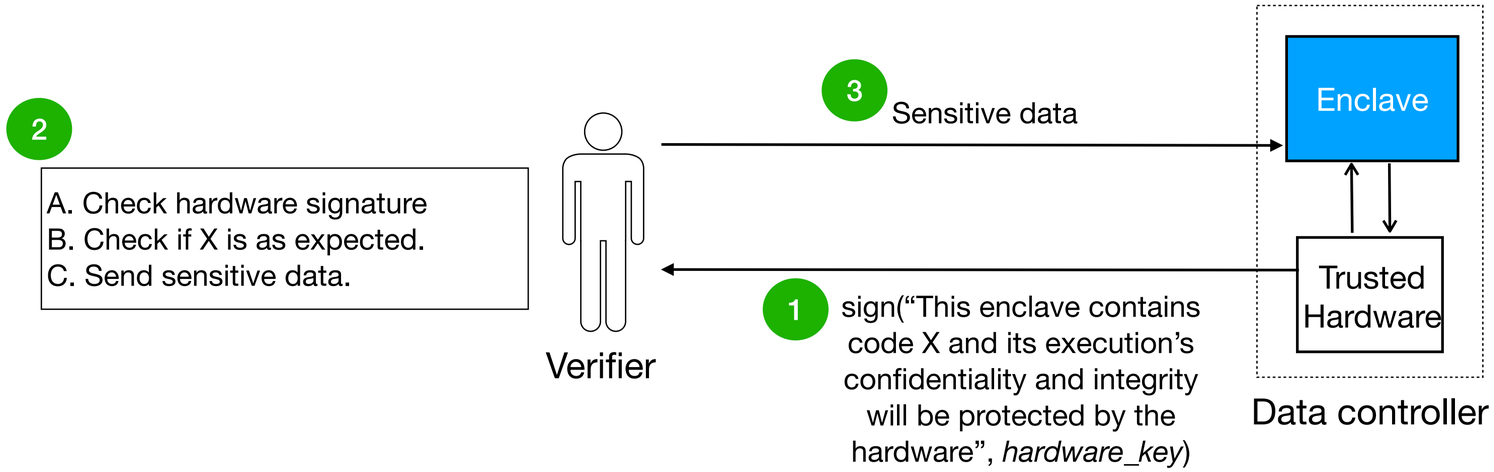
Careful about side-channel attacks though!

Confidentiality: *No one can peek into any state of the program running within the enclave.*

Integrity: *No one can modify execution of the program running within the enclave, except through explicit entry points.*

Remote attestation: *Sensitive data will only be processed as per code X.*





Encrypted DBs

Practical secure remote execution requires interacting with encrypted DBs

- **Querying encrypted databases:**
  - Searchable encryption schemes, searchable indices along with encrypted data (susceptible to reconstruction attacks)
  - EnclaveDB: entire database within an enclave

- **Private information retrieval:** only hides access patterns while DB is in cleartext (perhaps not very useful for us)

Summary

- Privacy requires a multi-pronged approach and many techniques exist for each, but an overarching privacy architecture is missing
- Purpose limitation is a crucial privacy requirement, considering the impossibility of absolute privacy
- Almost no technique talks about an external regulator preventing illegal access and processing of data (tomorrow’s talk)
- Issues related to consent and privacy self-management (tomorrow’s talk)

References

- Rivest et al. ’78: *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,* CACM, Vol 21 No. 2, 1978
- Diffie & Hellman ’76: *New Directions in Cryptography,* IEEE Tr. Information Theory, Vol 22 No. 6, 1976
- Goldwasser et al. ’89: *The Knowledge Complexity of Interactive Proof Systems,* SIAM J. Computing, Vol 18 pp. 186-208, 1989
- Goldreich et al. ’91: *Proofs That Yield Nothing but Their Validity or All Languages in NP Have Zero-knowledge Proof Systems,* J. ACM, Vol 38 pp. 690-728, 1991
- Pedersen ’91: *Non-interactive and information-theoretic secure verifiable secret sharing,* CRYPTO, pp. 129-140, 1991
- Fiat & Shamir ’86: *How To Prove Yourself: Practical Solutions to Identification and Signature Problems,* CRYPTO, pp. 186-194, 1986
- Chaum ’85: *Security without Identification*, CACM, Vol 28 No. 10, 1985
- Narayanan et al. ’19: *Robust de-anonymization of large sparse datasets: a decade later (https://www.cs.princeton.edu/~arvindn/publications/de-anonymization-retrospective.pdf)*
- Dinur & Nissim ’03: *Revealing information while preserving privacy*, PODS, 2003
- Dwork ’05: *Differential privacy*, ICALP, 2006