Secure Multi-party Computation

What it is, and why you'd care

Manoj Prabhakaran

University of Illinois, Urbana-Champaign

SMC

SMC

- SMC conceived more than 30 years back

SMC

- SMC conceived more than 30 years back
- A very general concept that subsumes the bulk of theoretical cryptography

SMC

- SMC conceived more than 30 years back
- A very general concept that subsumes the bulk of theoretical cryptography
- Largely a well-kept secret

SMC: the question

SMC: the question

- Collaboration without trust?

SMC: the question

- Collaboration without trust?
  - Collaboration: compute on collective data belonging to different parties

SMC: the question

- Collaboration without trust?
  - Collaboration: compute on collective data belonging to different parties
    - e.g. query with me, database with you

SMC: the question

- Collaboration without trust?
  - Collaboration: compute on collective data belonging to different parties
    - e.g. query with me, database with you
    - e.g. query with me, encrypted database with you, key with someone else

SMC: the question

- Collaboration without trust?
  - Collaboration: compute on collective data belonging to different parties
    - e.g. query with me, database with you
    - e.g. query with me, encrypted database with you, key with someone else
  - Goal: Nothing should be revealed “beyond the result”

SMC: the question

- Collaboration without trust?
  - Collaboration: compute on collective data belonging to different parties
    - e.g. query with me, database with you
    - e.g. query with me, encrypted database with you, key with someone else
  - Goal: Nothing should be revealed “beyond the result”
  - “Ideally”: Use a trusted third party

SMC: the question

- Collaboration without trust?
  - Collaboration: compute on collective data belonging to different parties
    - e.g. query with me, database with you
    - e.g. query with me, encrypted database with you, key with someone else
  - Goal: Nothing should be revealed “beyond the result”
  - “Ideally”: Use a trusted third party
  - “Really”: Can’t agree on a trusted party. So...

SMC: the answer

SMC: the answer

- SMC protocol: among mutually distrusting parties, to emulate the presence of a globally trusted party

SMC: the answer

- SMC protocol: among mutually distrusting parties, to emulate the presence of a globally trusted party
- Numerous protocols in literature for various functionalities, in various settings

SMC: the answer

- SMC protocol: among mutually distrusting parties, to emulate the presence of a globally trusted party
- Numerous protocols in literature for various functionalities, in various settings
  - Tools: Verifiable secret-sharing, homomorphic encryptions, commitments, ZK proofs, oblivious transfer, …

SMC: the answer

- SMC protocol: among mutually distrusting parties, to emulate the presence of a globally trusted party
- Numerous protocols in literature for various functionalities, in various settings
  - Tools: Verifiable secret-sharing, homomorphic encryptions, commitments, ZK proofs, oblivious transfer, …
- Simpler protocols if some trust already present

SMC: the answer

- SMC protocol: among mutually distrusting parties, to emulate the presence of a globally trusted party
- Numerous protocols in literature for various functionalities, in various settings
  - Tools: Verifiable secret-sharing, homomorphic encryptions, commitments, ZK proofs, oblivious transfer, …
- Simpler protocols if some trust already present
  - “Honest-but-curious”

SMC: the answer

- SMC protocol: among mutually distrusting parties, to emulate the presence of a globally trusted party
- Numerous protocols in literature for various functionalities, in various settings
  - Tools: Verifiable secret-sharing, homomorphic encryptions, commitments, ZK proofs, oblivious transfer, …
- Simpler protocols if some trust already present
  - “Honest-but-curious”
  - “Honest-majority”

SMC: the answer

- SMC protocol: among mutually distrusting parties, to emulate the presence of a globally trusted party
- Numerous protocols in literature for various functionalities, in various settings
  - Tools: Verifiable secret-sharing, homomorphic encryptions, commitments, ZK proofs, oblivious transfer, …
- Simpler protocols if some trust already present
  - “Honest-but-curious”
  - “Honest-majority”
  - Simple (offline) trusted sources

SMC in GENI?

SMC in GENI?

- Where privacy is needed

SMC in GENI?

- Where privacy is needed
  - e.g. Measurement archives held by a *virtual* trusted party

SMC in GENI?

- Where privacy is needed
  - e.g. Measurement archives held by a *virtual* trusted party
  - Secure distributed storage and computation (secure unless all servers corrupt)

SMC in GENI?

- Where privacy is needed
  - e.g. Measurement archives held by a *virtual* trusted party
  - Secure distributed storage and computation (secure unless all servers corrupt)
  - May use “honest majority” in a federation

SMC in GENI?

- Where privacy is needed
  - e.g. Measurement archives held by a *virtual* trusted party
  - Secure distributed storage and computation (secure unless all servers corrupt)
  - May use “honest majority” in a federation
- Provide SMC as an “experiment support service”?

SMC in GENI?

- Where privacy is needed
  - e.g. Measurement archives held by a *virtual* trusted party
  - Secure distributed storage and computation (secure unless all servers corrupt)
  - May use “honest majority” in a federation
- Provide SMC as an “experiment support service”?
  - SMC offers a whole range of novel applications