Advances in Secure Multiparty Computation and Suggestions for Further Research

W. KONRAD VESEY

ELKRIDGE SECURITY

konrad@elkridgesecurity.com

Abstract

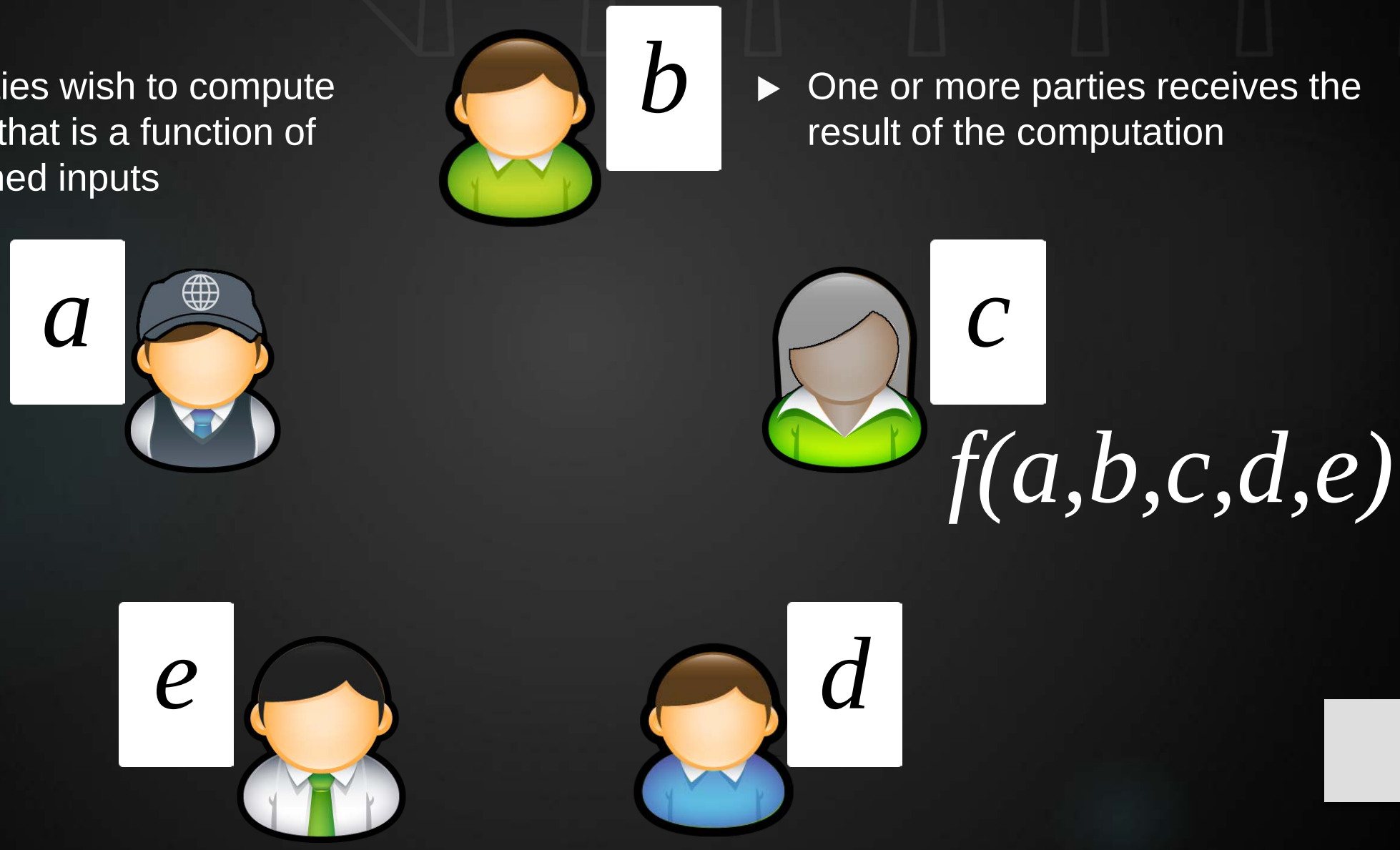
A growing number of tasks in modern life involve processing a combination of private personal information and information from public or commercial sources. This requires us to share private information with more and more organizations and businesses and hope that the combined exposure does not result in disclosures that would harm us. This is also a concern when we share private information with government agencies in order to satisfy legal requirements or to receive services. The amount and diversity of the private information we share with others simply to participate in civil society is growing and the risks are growing as well with few tools at hand to mitigate those risks. A question that once would have seemed absurd can and must now be seriously considered: is it necessary to share private information in order to process it in combination with other data? Relatedly, is it necessary for the government to have access to private information to meet its legitimate operational needs?

For the last thirty years, techniques have been studied to allow computations to take place on data while it is encrypted. Crucially for our present needs, these techniques allow computations to be performed securely on information provided by multiple parties, each of whom wants its input kept private from all other parties. This is the field of secure multiparty computation (MPC), a research area that has seen rapid improvements in technology efficiency in recent years.

This presentation will describe some variations on the secure MPC problem that were explored by the Security and Privacy Assurance Research (SPAR) program at the Intelligence Advanced Research Projects Activity (IARPA), discuss some lessons learned, and suggest some areas for further research that could enhance secure MPC as a broadly applicable tool for privacy protections.

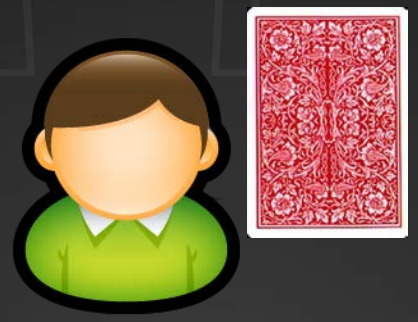
Secure Multiparty Computation (MPC)

- A set of parties wish to compute information that is a function of their combined inputs
- One or more parties receives the result of the computation

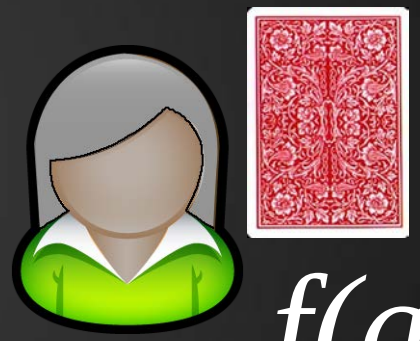


Secure Multiparty Computation (MPC)

- A set of parties wish to compute information that is a function of their combined inputs

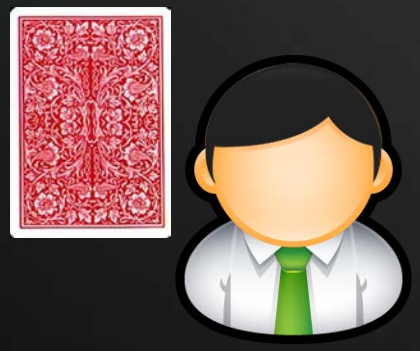


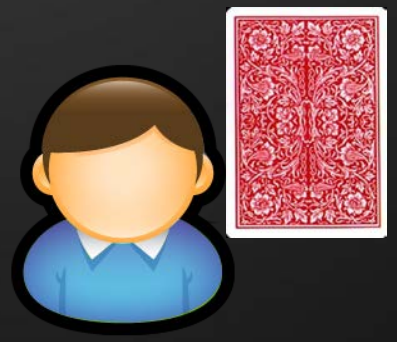
- One or more parties receives the result of the computation



$$f(a,b,c,d,e)$$

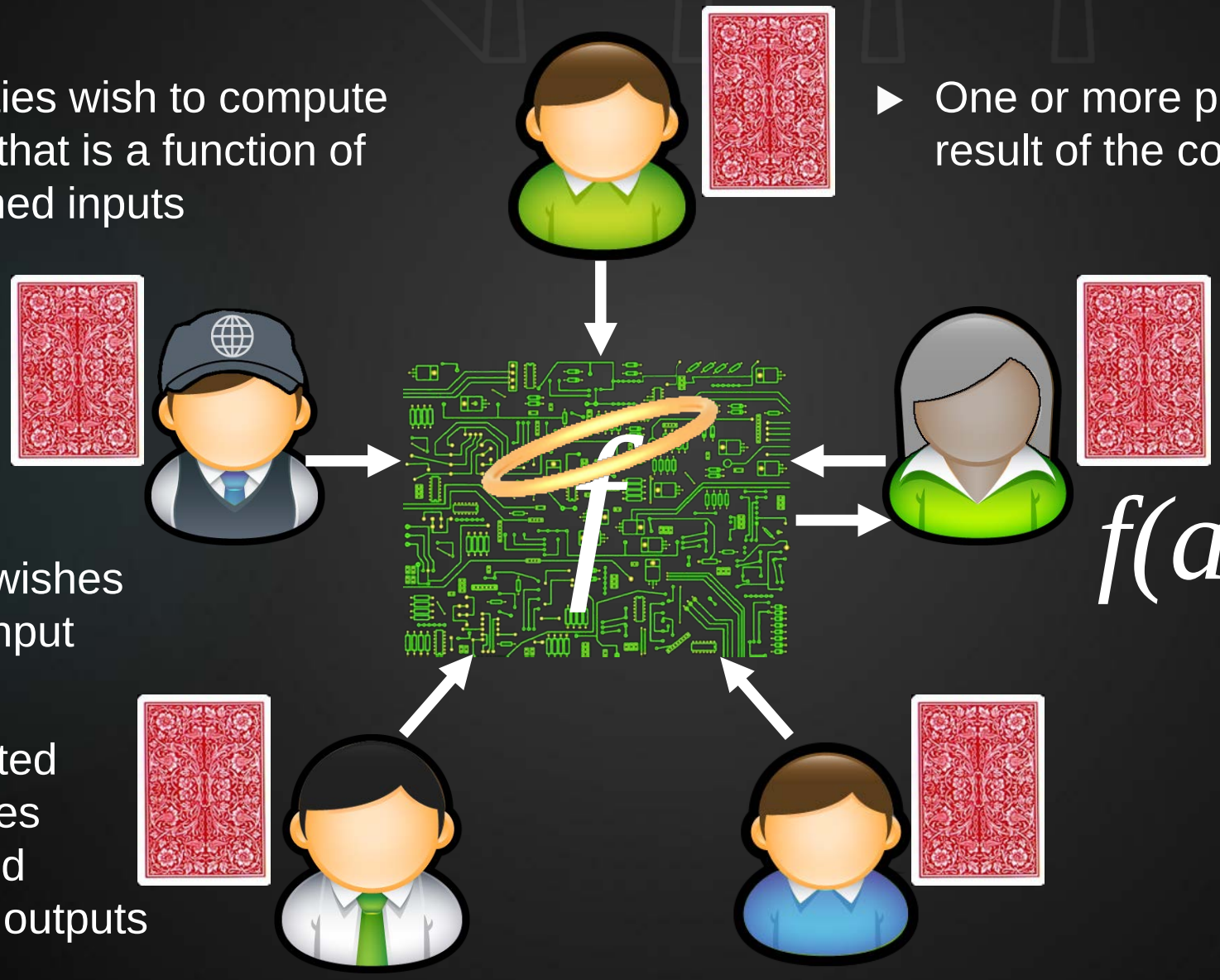
- Each party wishes to keep its input private





Secure Multiparty Computation (MPC)

- A set of parties wish to compute information that is a function of their combined inputs
- One or more parties receives the result of the computation

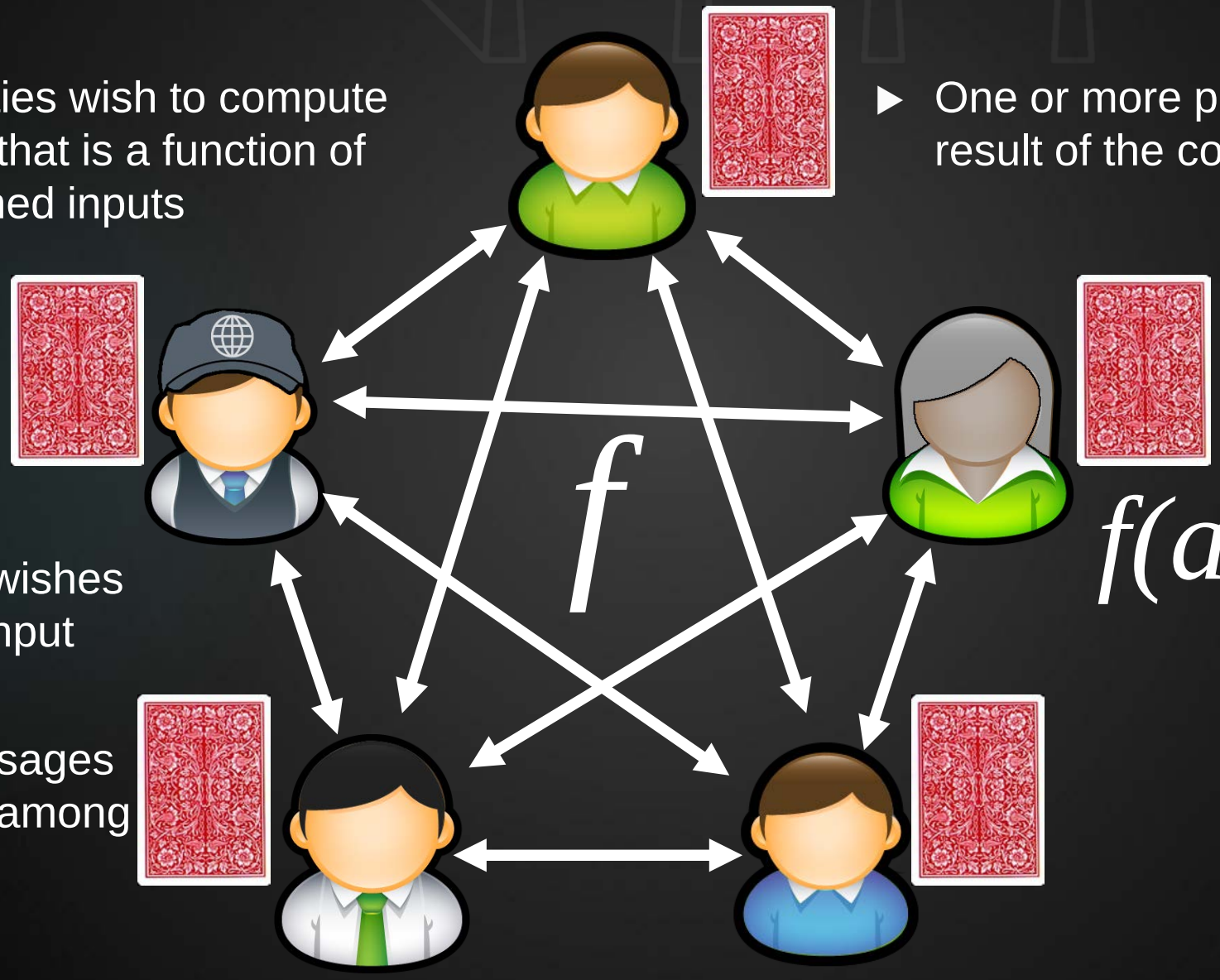


$f(a,b,c,d,e)$

- Each party wishes to keep its input private
- Ideal: A trusted party receives all inputs and delivers the outputs

Secure Multiparty Computation (MPC)

- A set of parties wish to compute information that is a function of their combined inputs
- One or more parties receives the result of the computation



$f(a,b,c,d,e)$

- Each party wishes to keep its input private
- Actual: Messages are passed among the parties

Secure MPC Application: Privacy-Preserving Data Retrieval

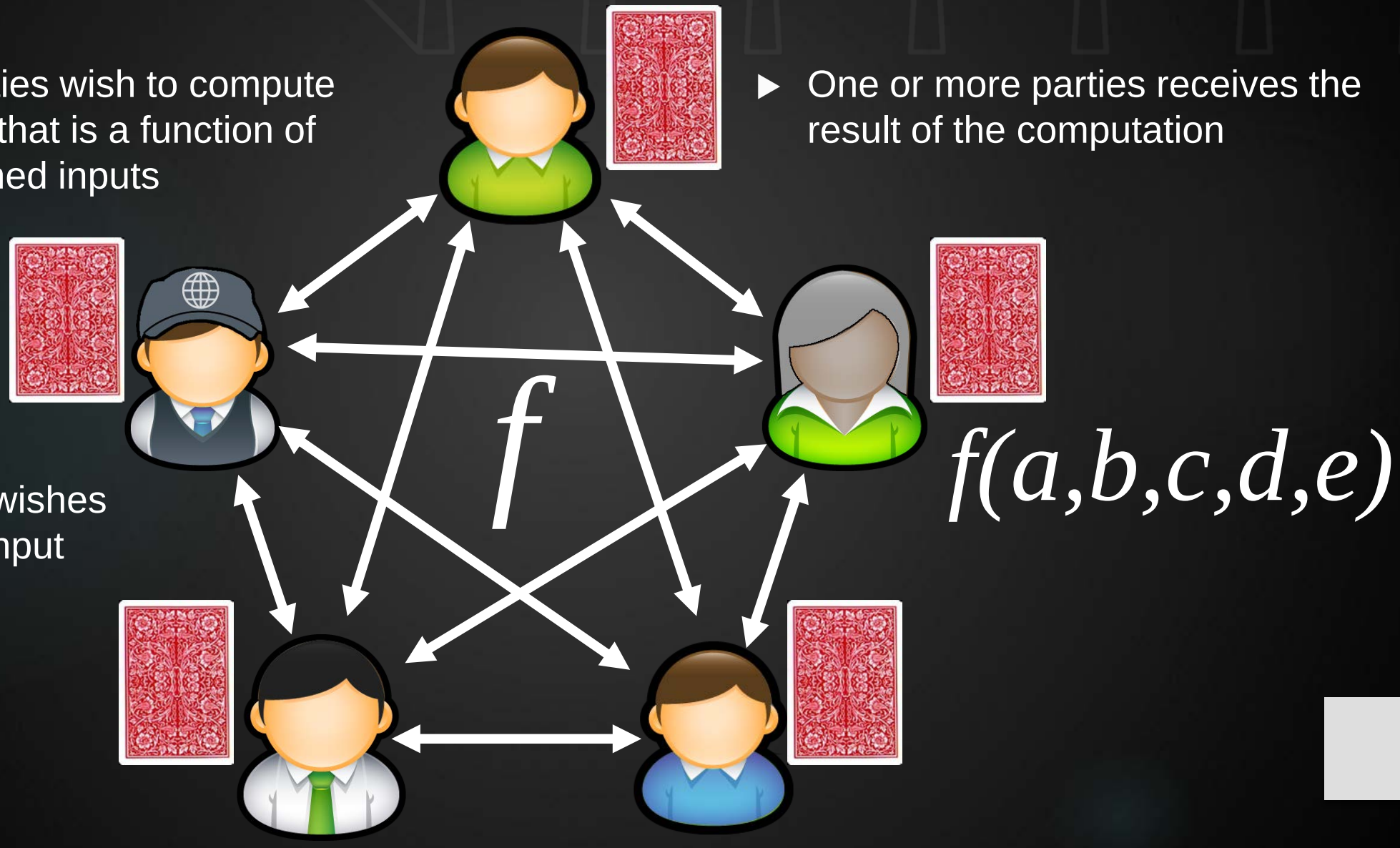
- Select from a data set without revealing which item was retrieved and without accessing any item that is not selected
  - IT vulnerability information
  - Health-related searches
  - Fraud/compliance investigations
  - Investment monitoring
  - Law enforcement investigations
  - Counterintelligence/Counterterrorism

Secure MPC Application: Privacy-Preserving Data Analysis

- Extract information from data without learning the data
  - Statistical analysis
  - Image/signal processing
  - Graph/netflow analysis
  - Virus scanning
  - Business valuation
  - Auctions
  - Biomedical research
  - Regional public health trends
  - Pattern matching
  - Data mining
  - Biometric identification
  - Fraud detection
  - Genome analysis
  - Price negotiation
  - Pharmacological research
  - Targeted advertising
- Multiple data sources can be combined in use that cannot be aggregated in storage
- MPC leaves no footprint of data that could be used for future unintended purposes

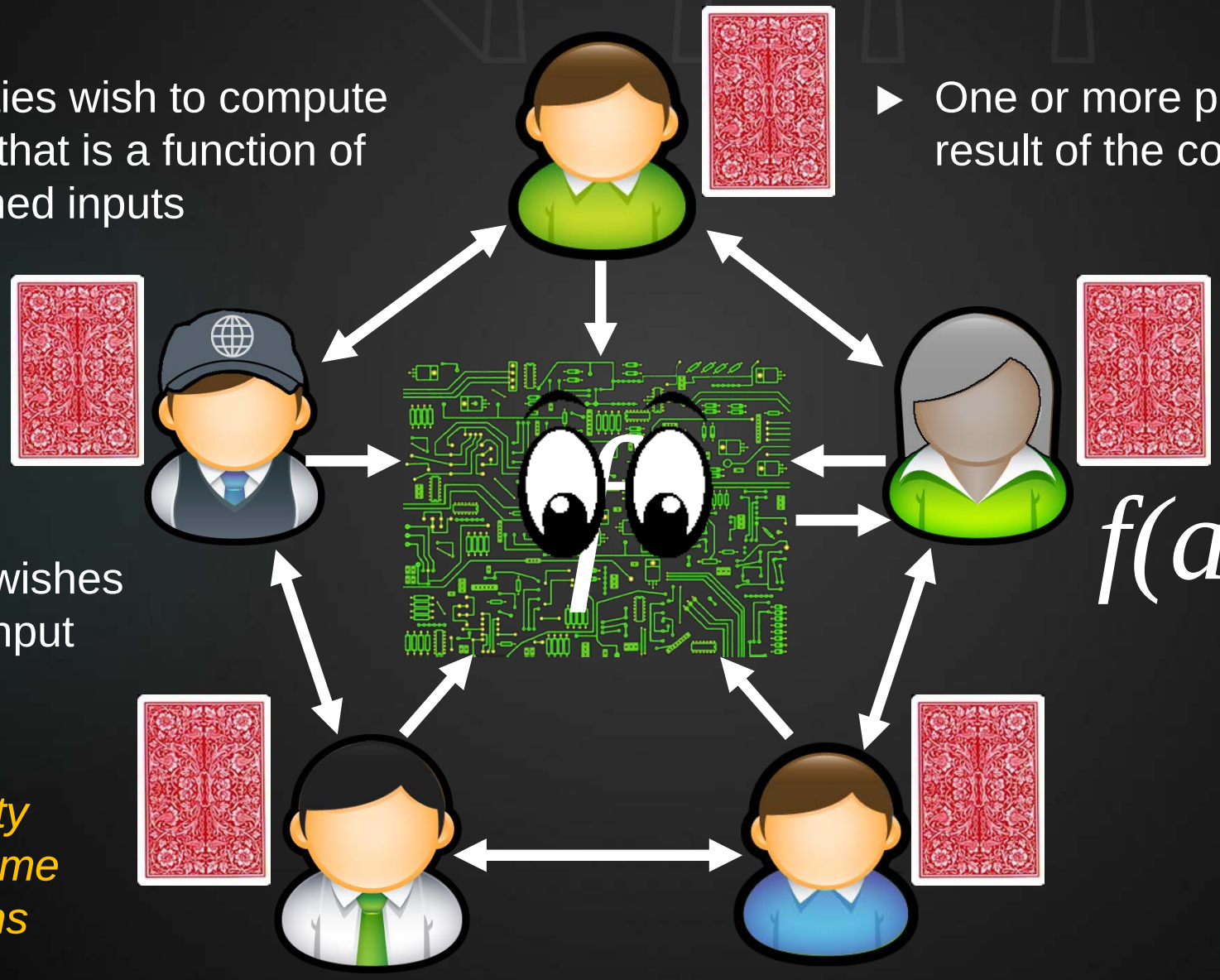
Secure Multiparty Computation (MPC)

- A set of parties wish to compute information that is a function of their combined inputs
- One or more parties receives the result of the computation
- Each party wishes to keep its input private



Secure Multiparty Computation (MPC)

- A set of parties wish to compute information that is a function of their combined inputs
- One or more parties receives the result of the computation

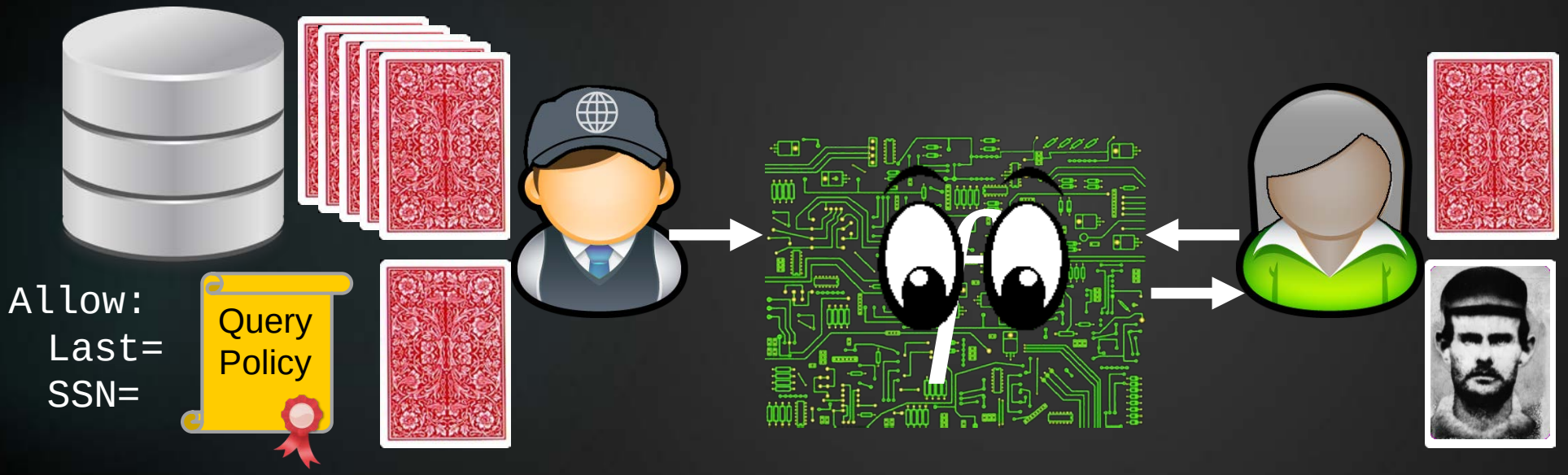


$f(a,b,c,d,e)$

- Each party wishes to keep its input private
- *Variation: A "helper" party performs some computations*

Security and Privacy Assurance Research

- One party holds a database of private U.S. person information and a query authorization policy
- One party holds a sensitive or classified query

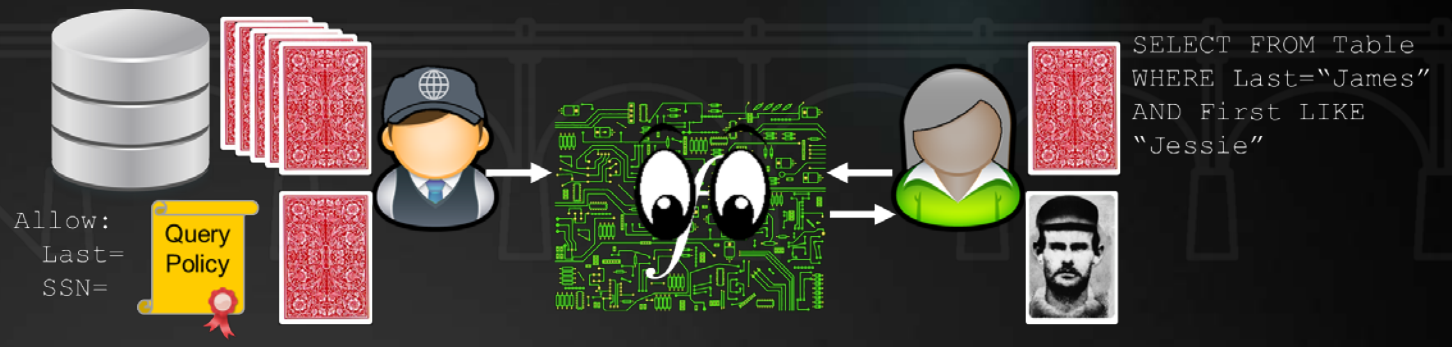



```
SELECT FROM Table
WHERE Last="James"
AND First LIKE
"Jessie"
```

- A record is revealed only if it matches an authorized query

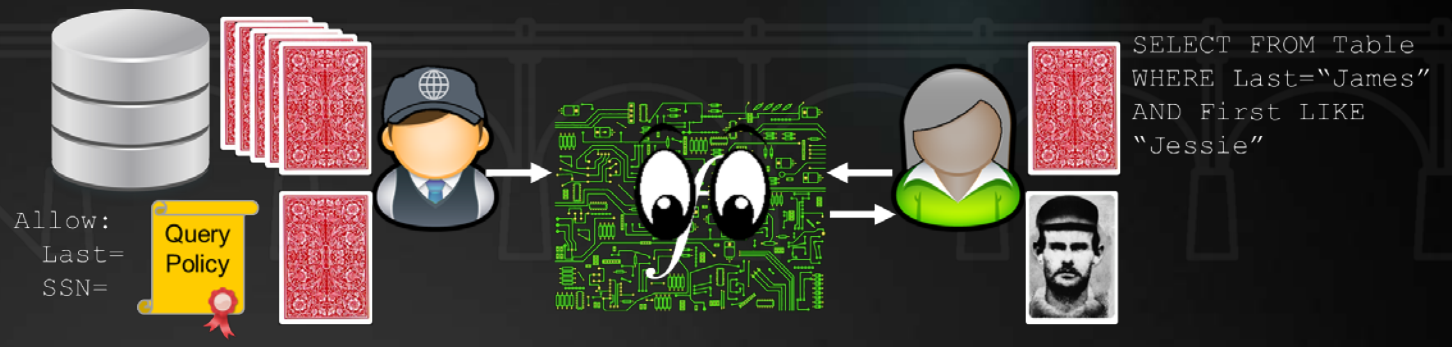
- Efficiency is gained by using a semi-trusted intermediate party who sees only encrypted data but may learn some "access pattern" information:
  - The type of query (exact match, fuzzy match, etc.), the attributes included (Last, First), the number of matching records, whether a record was a match for multiple queries, etc.

SPAR Lessons



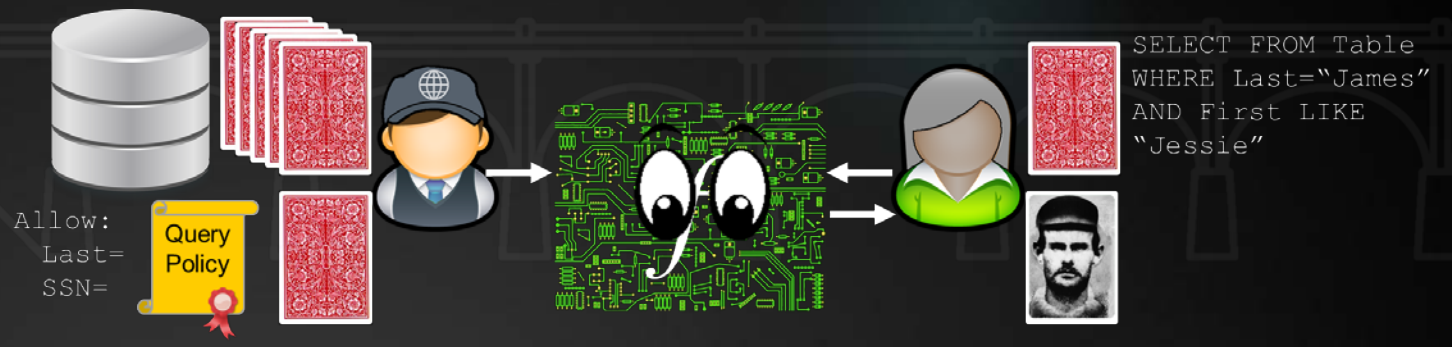
- Real-world applications are rarely "classic" MPC:
  - Structured inputs to a well-defined function yielding a structured output
- SPAR protocols are cryptography applied to a multifaceted distributed computing problem
  - Database access is a complex operation requiring secure efficient support for
    - Query expressions involving multiple types of varying complexity and size, submitted against
    - Databases with diverse schemata and sizes up to 10 Terabytes, while enforcing
    - Query authorization policy expressions of varying complexity and size, resulting in
    - Outputs that could be zero to millions of records in size
    - The performance is highly dependent on the relationship between query rate, network latency, memory size, and the division of effort in the protocol design between communication and computation
- *The future of MPC will involve a lot of SPAR-like applications*

SPAR Lessons



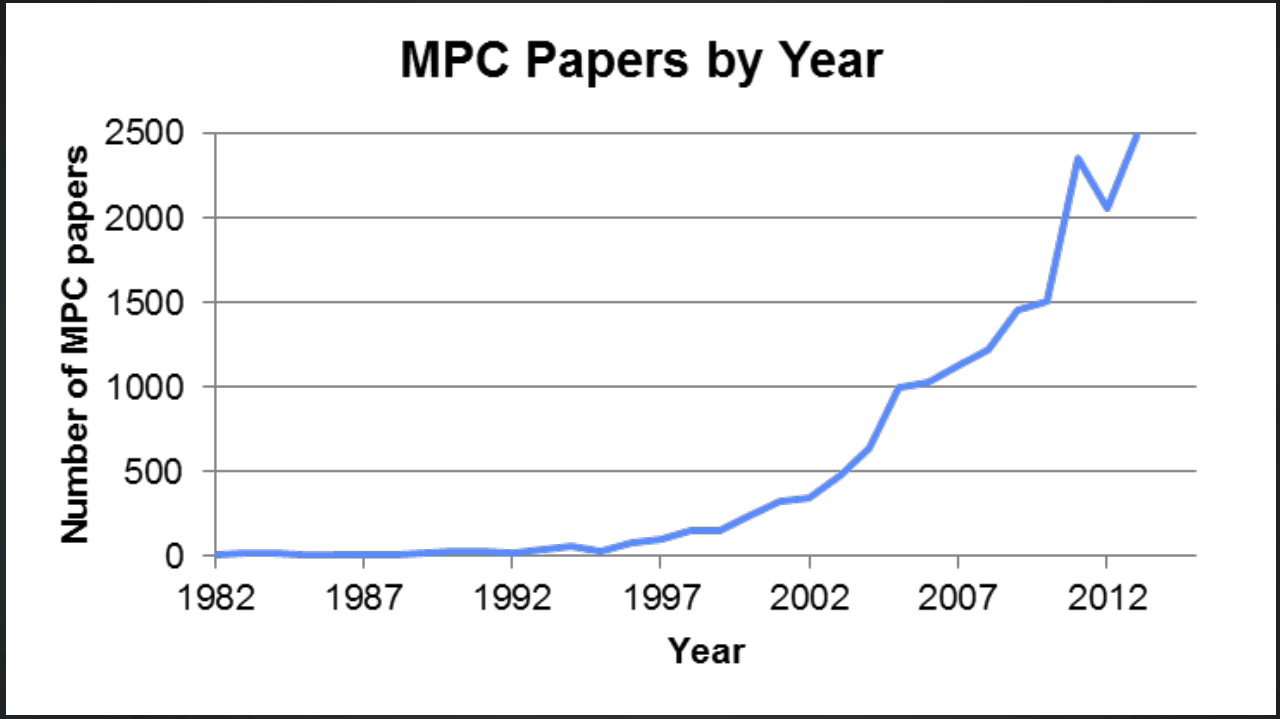
- Efficiency can be achieved by allowing access pattern leakage to the intermediate party
- *Many real-world applications can tolerate some access pattern leakage, especially if there are significant performance gains*

SPAR Lessons



- Protocol design is sensitive to "minor" differences in application usage characteristics
  - Key size, database record size, frequency of protocol execution, entropy of the underlying data, types of queries made in practice, types of query policies used in practice, tolerance for specific types of access pattern information leakage, etc.
  - *These are not "minor" differences to a cryptographer, but are minor differences to the application end user*
  - Application usage characteristics seem minor to the end user because they don't change the essential character of the application:
    - What data are being exchanged between the parties and for what purpose
    - The value of the data to each party and the fundamental protection requirements
  - *Usage characteristics drift over time as the application system goes through its mission relevance life cycle*

Recent Research in MPC



Recent Research in MPC

- Privacy-preserving data structures
  - Koller and Scholl, “Efficient, Oblivious Data Structures for Secure MPC,” 2014
- Efficient subprotocols/primitives
  - Wang, Chan, and Shi, “Circuit ORAM,” 2015, construction achieves a theoretical lower bound, solving a 27-year open problem
- Special-purpose computations
  - Hamada, Ikarashi, Chida, and Takahashi, “Oblivious Radix Sort: An Efficient Sorting Algorithm for Practical Secure Multi-party Computation,” 2014
  - Pullonen and Siim, “Combining Secret Sharing and Garbled Circuits for Efficient Private IEEE 754 Floating-Point Computations,” 2014

Recent Research in MPC

- Crowds
  - Said and Zamani, 2014, “Recent Results in Scalable Multi-Party Computation,”
  - Zamani, Mohavedi, and Saia, 2014, “Millions of Millionaires: Multiparty Computation in Large Networks” (MPC scaled to large numbers of participants)
- Low compute power
  - Carter, Lever, and Traynor, “Whitewash: Outsourcing Garbled Circuit Generation for Mobile Devices,” 2014
- Special delivery requirements
  - Kilinc and Kupcu, “Optimally Efficient Multi-Party Fair Exchange and Fair Secure Multi-Party Computation,” 2014
  - Cohen and Lindell, “Fairness Versus Guaranteed Output Delivery in Secure Multiparty Computation,” 2014

Recent Research in MPC

- Audit of correctness
  - Baum, Damgard, and Orlandi, “Publicly Auditable Secure Multi-Party Computation,” 2014
  - Schoenmakers and Veeningen, “Universally Verifiable Multiparty Computation from Threshold Homomorphic Cryptosystems,” 2015
- Proof generation
  - Pettai and Laud, “Automatic Proofs of Privacy of Secure Multi-Party Computation Protocols Against Active Adversaries,” 2014
- Protocol generation/compilers
  - Wysteria, FairplayMP, Sharemind, VIFF, Sepia, TASTY

Suggestion for Research

- **Research is needed to provide the tools and techniques to find efficient solutions for protecting security and privacy in complex MPC applications with large-scale data sets**
  - **Research is needed in protocol design tools to automatically compile efficient tailored MPC protocols from application specifications and libraries of efficient primitives**
    - **Research is needed to classify and reason about leakage to fully understand the impact of leakage/efficiency tradeoffs in MPC protocol design**
    - **Research is needed in new or more efficient cryptographic primitives as building blocks for complex secure and privacy protecting distributed computing applications**
  - **Research is needed in proof checking tools to rapidly verify the security and privacy protections of compiler-generated MPC protocol specifications**
  - **Research is needed in assured software engineering techniques to ensure that MPC protocol implementations are correct and robust**

Speaker Bio

Konrad Vesey has over 25 years of experience in cybersecurity, cryptography, software engineering, and information assurance. Mr. Vesey worked for the National Security Agency as a Cryptologic Mathematician, a Global Network Vulnerability Analyst, and as the Director of the Center for Assured Software. He was a Program Manager for the Intelligence Advanced Research Projects Activity and is a recognized expert on security and privacy. Mr. Vesey founded Elkridge Security in 2015. The opinions, conclusions, and recommendations are those of the speaker and are not necessarily endorsed by the U.S. Government.