On Diophantine Complexity and Statistical Zero-Knowledge Arguments

Helger Lipmaa
Helsinki University of Technology
`http://www.tcs.hut.fi/~helger`

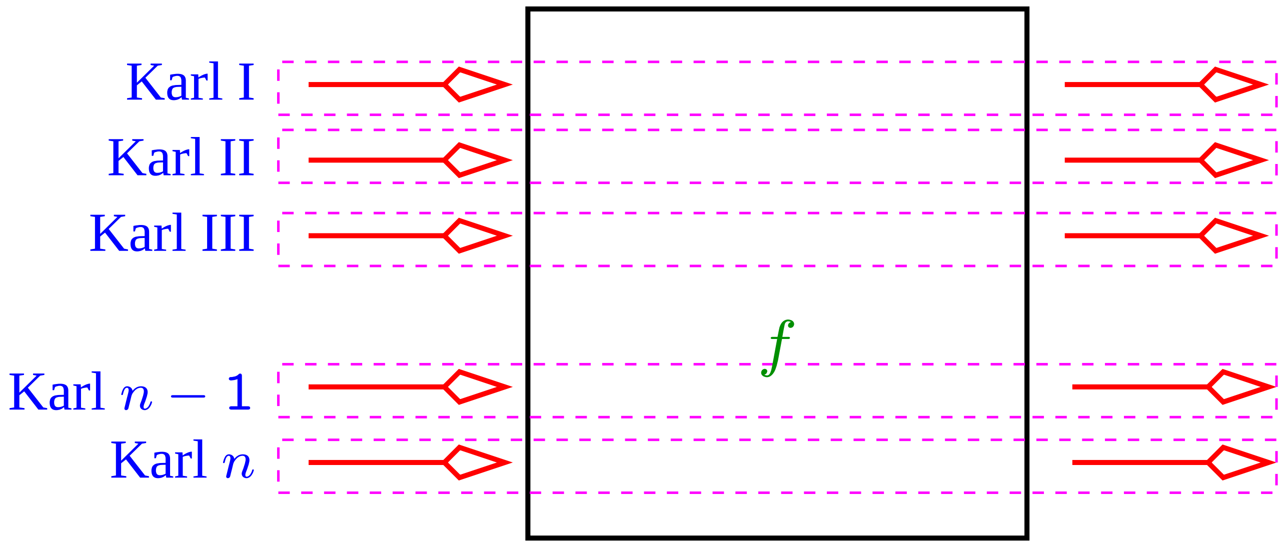
Overview of This Talk

- Cryptographic protocols, limitations
- Outsourcing model
- Polynomials and integer commitment schemes
- Efficient solutions by using diophantine complexity

Reminder: Multi-Party Computation

- All efficiently computable functions can also be computed securely
- Assume there are n participants, and the ith participant has input x_i. Assume f is a function $f(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$.
- There is a way (*multi-party computation*) to compute f so that at the end of the protocol, the ith participant will get the know value of y_i and nothing else, except what she could compute from (x_i, y_i) herself.

We Gotta Have Some Pictures



Assume f is any function. Karl's can compute f so that (a) Security: Karl i obtains the output he wanted to obtain, (b) Privacy: Karl i will not obtain any new information that cannot be computed from his input and output alone.

Applications: Voting

- n voters, one tallier.
- Voter i has input v_i, her vote.
- Security: Tallier gets to know $y_T := \sum_{i=1}^{n} v_i$.
- Privacy: Tallier will not get any information that cannot be computed from y_T alone. Voters will not get any new information at all.

Limitations

- MPC: To get total privacy and security, a majority of the parties must be honest (in some settings, $2/3$!)

- “Threshold trust” in voting: assume that a majority of talliers and/or voters is honest?

- Two-party computation: privacy possible, but security is possible only for one of the two parties (since he can halt as soon as he recovers his output)

- Fortunately, often one can design protocols, where halting is not a problem — but not always

Outsourcing model

- n individuals, 1 interested third party S, one established authority A.
- Individual i has input v_i, her financial or social choice (vote, bid, ...).
- Security: S gets to know $y_T := f(v_1, \ldots, v_n)$ for some destination function f.
- Privacy: S will not get any information that cannot be computed from y_T alone. Individuals will not get any new information at all. A can get to know the vector $(v_1, \ldots, v_n)$.

Why makes sense?

- In voting, it is better to have one tallier: in real life, very hard to have a multiple of completely independent talliers.

- Same in auctions: there is a single seller, all servers are operated by him; why should we trust m machines controlled by the same person more than just one machine, controlled by him?

- OTOH: A can be an established authority who has a reputation to take care off; often S is an occassional party.

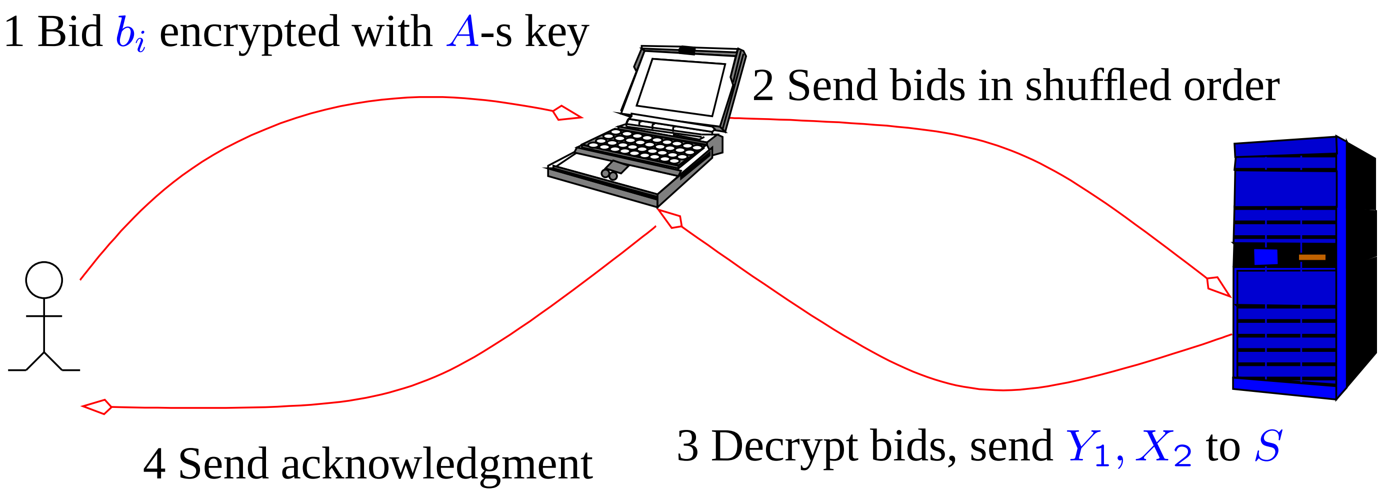
- It is also possible to design the system so that we can avoid the limitations of the two-party and multi-party computations, *efficiently*

Example: Vickrey Auctions

Security requirements:

- Correctness
  - ⋆ Highest bidder Y_1 should win
  - ⋆ He should pay the second highest bid X_2
- Privacy: S should not get any information about the bids but (Y_1, X_2)
- Scheme should be secure unless both A and S are malicious

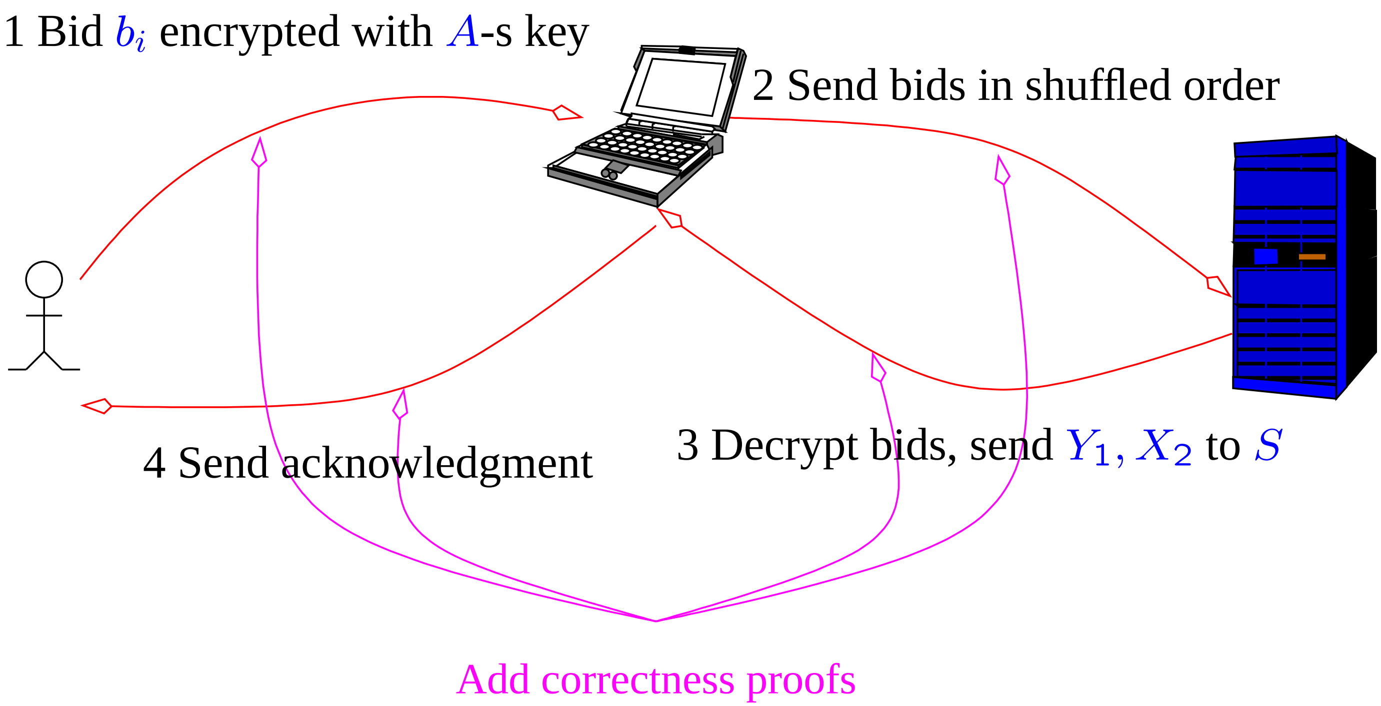
Simple scheme



S will not get any extra information, but S can increase X_2

$A \longrightarrow S$ interaction is quite large

Simple scheme $\longrightarrow$ complex scheme



Proofs of correctness

1. Complex: use bulletin board, argue that bid belongs to some set
2. Complex: combine bids, argue correctness of combination
3. Complex: extract X_2, argue it
4. Simple: (Y_1, X_2) signed by S

Efficient Proofs of Knowledge

1. Bidders encode their bids by using some function $\mathsf{enc}(\cdot)$, and then encrypt the result by using A's key. They send the result, $E_K(\mathsf{enc}(b_i); r_i)$ to S

2. S multiplies the results, gets $E_K(\sum \mathsf{enc}(b_i); \sum r_i)$; sends the result to A

3. A decrypts the result, obtains $\sum \mathsf{enc}(b_i)$, applies a decoding function to it and obtains $(b_1, \dots, b_n)$

4. A computes $o = f(b_1, \dots, b_n)$, sends this to S and argues that o was correctly computed

Details!

1. E is homomorphic: $E_K(m_1; r_1)E(m_2; r_2) = E_K(m_1 + m_2; r_1 + r_2)$ — such E are well-known (Paillier, ...)

2. There exists $\mathsf{enc}(\cdot)$ and $\mathsf{dec}(\cdot)$, such that $\mathsf{dec}(\sum \mathsf{enc}(b_i)) = (b_1, \dots, b_n)$ for all b_1 from $[0, V-1]$ — for example, take $\mathsf{enc}(b_i) = V^{b_i}$; then $\mathsf{dec}(b)$ returns the vector of V-radix positions of b

3. Thus a bidder must argue that c_i is an encryption of V^{b_i} for $b_i \in [0, V-1]$, and A must argue that $o = f(\mathsf{dec}(\sum \mathsf{enc}(b_i))$

Problems!

1. Known arguments that $c_i = E_K(V^\mu; \rho) \wedge \mu \in [0, V-1]$ are long [DJ01,LAN02]

2. Efficient arguments for $o = f(\mathsf{dec}(\sum \mathsf{enc}(b_i))$ are known only for a very limited set of f-s

3. For example, in Vickrey auctions one needs to argue that $c = E_K(\mu; \rho) \wedge \mu \in [0, V-1]$; even for this range argument, conventional arguments are too long.

Integer commitment schemes

- Commitment scheme: $c = C_K(\mu; \rho)$. Hiding: c does not give any information about μ. Binding: hard to find $\mu' \neq \mu$ such that $C_K(\mu; \rho) = C_K(\mu'; \rho')$.

- Integer: usually $\mu' \neq \mu$ means $\mu' \neq \mu \mod n$ for some finite n. In an integer commitment scheme, $\mu' \neq \mu$ is taken over integers.

Integer commitment schemes

- Homomorphic:

$$C_K(\mu_1 + \mu_2; \rho_1 + \rho_2) = C_K(\mu_1 + \mu_2; \rho_1) C_K(\mu_1 + \mu_2; \rho_2)$$

- Easy to argue that

$$c_1 = C_K(\mu_1; \cdot) \wedge c_2 = C_K(\mu_2; \cdot) \wedge c_3 = C_K(\mu_1 \mu_2; \cdot)$$

  this generalizes to an argument

$$c_1 = C_K(\mu_1; \cdot) \wedge c_2 = C_K(\mu_2; \cdot) \wedge c_3 = C_K(f(\mu_1, \mu_2); \cdot)$$

  for for every $f \in \mathbb{Z}[X]$

Diophantine Arguments

- Example: how to prove that $c = C_K(\mu; \cdot) \wedge \mu \geq 0$: by Lagrange, $\mu \geq 0 \iff (\exists_b \omega_1, \omega_2, \omega_3, \omega_4)[\mu = \omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2]$

- Generally: demonstrate that you know ω, such that $f(\mu; \omega) = 0$

Diophantine Arguments

1. Given μ, find such ω_i (Algorithm: Rabin-Shallit, slightly improved by us)

2. Commit to all ω_i, $c_i = C_K(\omega_i; \rho_i)$

3. Argue in ZK that

$$c = C_K(\mu; \rho) \wedge (\wedge c_i = C_K(\omega_i; \rho_i)) \wedge f(\mu; \omega) = 0$$

where $f(\mu; \omega) = \mu - \sum \omega_i^2$

Diophantine Sets

- We want to prove that $\mu \in S$ for some language S. By results of Matiyasevich etc, there exists an $R_S \in \mathbb{Z}[X]$, s.t. $(\exists\omega)[R_S(\mu;\omega) = 0] \iff \mu \in S$

+ We need that one can compute ω efficiently if it exists

+ ω must be polynomially short (in $|\mu|$) when $\mu \in S$

- On the other hand, ω may exist even if $\mu \notin S$, but in this case it must be very long (nonpolynomially long)

- If such R_S exists we say $S \in \mathbf{PD}$

Main results

- For all languages S in bounded arithmetic, these requirements are satisfied. In particular, if $\mu \in S$ then $|\omega| \leq |\mu|^2$ while if $\mu \notin S$ then $|\omega| \geq 2^{|\mu|}$

- Bounded arithmetic includes most of the languages that are necessary in our application domain (auctions, voting etc)

- Our proof hinges on the efficient argument for exponential relationship, presented in the paper

- Finally, we show that if one takes $\mathtt{enc}(b_i) = Z_V(b_i)$ for certain Lucas sequence $Z_a(b)$, one can build more efficient arguments than in the case of exponentiation

Theorem Assume $\mu_1 > 1$, $\mu_3 > 0$ and $\mu_2 > 2$. The exponential relation $[\mu_3 = \mu_1^{\mu_2}]$ belongs to $\mathbf{PD}$. More precisely, let $E(\mu_1, \mu_2, \mu_3)$ be the next equation:

$$
\begin{aligned}
&[(\exists \omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6)(\exists_b \omega_7, \omega_8)] \\
&\quad [(\omega_2 = \omega_1\mu_1 - \mu_1^2 - 1) \wedge (\omega_2 - \mu_3 - 1 \geq 0) \wedge && (E1 - E2) \\
&\quad (\mu_3 - (\mu_1 - \omega_1)\omega_7 - \omega_8 = \omega_2\omega_3)) \wedge (\omega_1 - 2 \geq 0) \wedge && (E3 - E4) \\
&\quad ((\omega_1 - 2)^2 - (\mu_1 + 2)(\omega_1 - 2)\omega_5 - \omega_5^2 = 1) \wedge && (E5) \\
&\quad (\omega_1 - 2 = \mu_2 + \omega_6(\mu_1 + 2)) \wedge (\omega_7 \geq 0) \wedge (\omega_7 < \omega_8) \wedge && (E6 - E8) \\
&\quad (\omega_7^2 - \omega_1\omega_7\omega_8 - \omega_8^2 = 1) \wedge (\omega_7 = \mu_2 + \omega_4(\omega_1 - 2)] \ , && (E9 - E10)
\end{aligned}
$$

where '$\exists_b$" signifies a bounded quantifier in the following sense: if $\mu_3 = \mu_1^{\mu_2}$ then $E(\mu_1, \mu_2, \mu_3)$ is true with $|\omega| = \Theta(\mu_2^2 \log \mu_1) = o(|\mu|^2)$. On the other hand, if $\mu_3 \neq \mu_1^{\mu_2}$ then either $E(\mu_1, \mu_2, \mu_3)$ is false, or it is true but the intermediate witnesses ω_7 and ω_8 have length $\Omega(\mu_3 \log \mu_3)$, which is equal to $\Omega(2^{|}\mu| \cdot |\mu|)$ in the worst case.

Conclusions

- Argued for the outsourcing model for cryptographic protocols
- No threshold trust, efficient arguments of knowledge
- Showed that most of the necessary arguements in this model can be obtained efficiently by using integer commitment schemes
- New algorithm for Lagrange representation, new polynomial for the exponential relationship
- Idea of using Lucas sequences in the zero-knowlege arguments

Questions?

?