Cryptographic Protocols

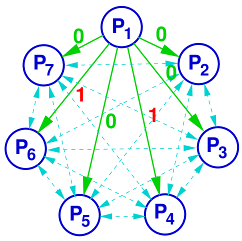
Spring 2019

Part 1

Cryptographic Protocols

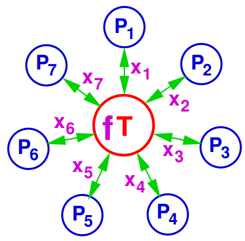
1. Interactive Proofs and Zero-Knowledge Protocols
   Proving without Showing
2. Secure Multi-Party Computation
   Computing without Knowing
3. Broadcast
   Agreeing without Trusting
4. Secure E-Voting

Broadcast / Byzantine Agreement



Theorem [LSP80]: Among n players, broadcast is achievable if and only if $t < n/3$ players are corrupted.

Secure Multi-Party Computation



Cryptographic Protocols

1. Interactive Proofs and Zero-Knowledge Protocols
2. Secure Multi-Party Computation
3. Broadcast

						4		
2					1		5	
4	3		7	5		1		2
				7			6	
	5	3				2	4	
	4			1				
3		1		8	2		7	4
	2		9					5
		8						

Formal Proofs (Conventional)

Proof system for a class of statements

- A statement (from the class) is a string (over a finite alphabet).
- The semantics defines which statements are true.
- A proof is a string.
- Verification function φ: (statement, proof) $\mapsto$ {accept, reject}.

Example: n is non-prime

- Statement: a number n (sequence of digits), e.g. „399800021“.
- Proof: a factor f, e.g. „19997“.
- Verification: Check whether f divides n.

Requirements for a Proof System

- **Soundness:** Only true statements have proofs.
- **Completeness:** Every true statement has a proof.
- **Efficient verifiability:** φ is efficiently computable.

Proof System: Sudoku has Solution

						4		
2					1		5	
4	3		7	5		1		2
				7			6	
	5	3				2	4	
	4			1				
3		1		8	2		7	4
	2		9					5
		8						

Good Proof System

- Statement: 9-by-9 Matrix $\mathcal{Z}$ over $\{1, \ldots, 9, \perp\}$.
- Proof: 9-by-9 Matrix $\mathcal{X}$ over $\{1, \ldots, 9\}$.
- Verification:
  1) ....
  2) ....

Stupid Proof System

- Statement: 9-by-9 Matrix $\mathcal{Z}$ over $\{1, \ldots, 9, \perp\}$.
- Proof: "" (empty string)
- Verification: For all possible $\mathcal{X}$, check if $\mathcal{X}$ is solution for $\mathcal{Z}$.

$\rightarrow$ **This is not a proof!**

Efficient Primality Proof

An efficiently verifiable proof that n is prime:

0. For small n (i.e., $n \leq T$), do table look-up (empty proof).
1. The list of distinct prime factors $p_1, \ldots, p_k$ of $n-1$.
   ($n - 1 = \Pi_{i=1}^{k} p_i^{\alpha_i}$)
2. Number a such that
   $$a^{n-1} \equiv 1 \ (\text{mod } n)$$
   and
   $$a^{(n-1)/p_i} \not\equiv 1 \ (\text{mod } n)$$
   for $1 \leq i \leq k$.
3. Primality proofs for $p_1, \ldots, p_k$ (recursion!).

Two Types of Proofs

Proofs of Statements:

- Sudoku $\mathcal{Z}$ has a solution $\mathcal{X}$.
- z is a square modulo m, i.e. $\exists x \ \ z = x^2 \pmod{m}$.
- The graphs $\mathcal{G}_0$ and $\mathcal{G}_1$ are isomorphic.
- The graphs $\mathcal{G}_0$ and $\mathcal{G}_1$ are non-isomorphic.
- P $=$ NP

Proofs of Knowledge:

- I know a solution $\mathcal{X}$ of Sudoku $\mathcal{Z}$.
- I know a value x such that $z = x^2 \pmod{m}$.
- I know an isomorphism π from $\mathcal{G}_0$ to $\mathcal{G}_1$.
- I know a non-isomorphism between $\mathcal{G}_0$ and $\mathcal{G}_1$????
- I know a proof for either P $=$ NP or P $\neq$ NP.
- I know x such that $z = g^x$.

Often: Proof of knowledge $\rightarrow$ Proof of statement (knowledge exists)

Static Proofs vs. Interactive Proofs

Static Proof

Prover P
knows statement s,
proof p

$\xrightarrow{p}$

Verifier V
knows statement s
$(s, p) \rightarrow$ {accept, reject}

Interactive Proof

Prover P
knows statement s,

$\xrightarrow{m_1}$

$\xleftarrow{m_2}$

$\ldots$

$\xrightarrow{m_\ell}$

Verifier V
knows statement s
$(s, m_1, \ldots, m_\ell) \rightarrow$ {accept, reject}

Motivation for IP's:

1. zero knowledge
2. more powerful
3. applications

Interactive Proofs: Requirements

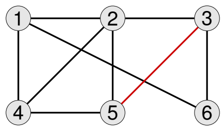
- **Completeness:** If the statement is true [resp., the prover knows the claimed information], then the correct verifier will always accept the proof by the correct prover.
- **Soundness:** If the statement is false [resp., the prover does not know the claimed information], then the correct verifier will accept the proof only with negligible probability, independent of the prover's strategy.

Desired Property:

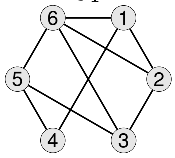
- **Zero-Knowledge:** As long as the prover follows the protocol, the verifier learns nothing but the fact that the statement is true [resp., that the prover knows the claimed information].

The Graph Isomorphism (GI) Problem

$\mathcal{G}_0$



$\mathcal{G}_1$



$$\begin{bmatrix} 0&1&0&1&0&1\\ 1&0&1&1&1&0\\ 0&1&0&0&\color{red}{1}&1\\ 1&1&0&0&1&0\\ 0&1&\color{red}{1}&1&0&0\\ 1&0&1&0&0&0 \end{bmatrix} \qquad \begin{bmatrix} 0&1&0&1&0&1\\ 1&0&1&0&0&1\\ 0&1&0&0&1&1\\ 1&0&0&0&1&0\\ 0&0&1&1&0&1\\ 1&1&1&0&1&0 \end{bmatrix}$$

Graph Isomorphism – One Round of the Protocol

Setting: Given two graphs $\mathcal{G}_0$ and $\mathcal{G}_1$.
Goal: Prove that $\mathcal{G}_0$ and $\mathcal{G}_1$ are isomorphic.

Peggy		Vic
knows $\mathcal{G}_0$, $\mathcal{G}_1$, σ s.t. $\mathcal{G}_1 = \sigma\mathcal{G}_0\sigma^{-1}$		knows $\mathcal{G}_0$ and $\mathcal{G}_1$
pick random permutation π $\mathcal{T} = \pi\mathcal{G}_0\pi^{-1}$	$\xrightarrow{\mathcal{T}}$	
	$\xleftarrow{c}$	$c \in_R \{0,1\}$
$c = 0 : \rho = \pi$ $c = 1 : \rho = \pi\sigma^{-1}$	$\xrightarrow{\rho}$	$c = 0 : \mathcal{T} \stackrel{?}{=} \rho\mathcal{G}_0\rho^{-1}$ $c = 1 : \mathcal{T} \stackrel{?}{=} \rho\mathcal{G}_1\rho^{-1}$

Graph-NON-Isomorphism – One Round of the Protocol

Setting: Given two graphs $\mathcal{G}_0$ and $\mathcal{G}_1$.
Goal: Prove that $\mathcal{G}_0$ and $\mathcal{G}_1$ are *not* isomorphic.

Peggy		Vic
knows $\mathcal{G}_0$ and $\mathcal{G}_1$		knows $\mathcal{G}_0$ and $\mathcal{G}_1$
		$b \in_R \{0,1\}$, π at random
	$\xleftarrow{\mathcal{T}}$	$\mathcal{T} = \pi\mathcal{G}_b\pi^{-1}$
if $\mathcal{T} \sim \mathcal{G}_0$: $r = 0$, if $\mathcal{T} \sim \mathcal{G}_1$: $r = 1$	$\xrightarrow{r}$	$r \stackrel{?}{=} b$

Fiat-Shamir – One Round of the Protocol

Setting: m is an RSA-Modulus.
Goal: Prove knowledge of a square root x of a given $z \in \mathbb{Z}_m^*$.

Peggy		Vic
knows x s.t. $x^2 = z \pmod{m}$		knows z
$k \in_R \mathbb{Z}_m^*$, $t = k^2$	$\xrightarrow{t}$	
	$\xleftarrow{c}$	$c \in_R \{0,1\}$
$r = k \cdot x^c$	$\xrightarrow{r}$	$r^2 \stackrel{?}{=} t \cdot z^c$

Guillou-Quisquater – One Round of the Protocol

Setting: m is an RSA-Modulus.
Goal: Prove knowledge of an e-th root x of a given $z \in \mathbb{Z}_m^*$.

Peggy		Vic
knows x s.t. $x^e = z \pmod{m}$		knows z
$k \in_R \mathbb{Z}_m^*$, $t = k^e$	$\xrightarrow{t}$	
	$\xleftarrow{c}$	$c \in_R \mathcal{C} \subseteq \{0,\ldots,e-1\}$
$r = k \cdot x^c$	$\xrightarrow{r}$	$r^e \stackrel{?}{=} t \cdot z^c$

Schnorr – One Round of the Protocol

Setting: Cyclic group $H = \langle h \rangle$, $|H| = q$ prime.
Goal: Prove knowledge of the discrete logarithm x of a given $z \in H$.

Peggy		Vic
knows $x \in \mathbb{Z}_q$ s.t. $h^x = z$		knows z
$k \in_R \mathbb{Z}_q$, $t = h^k$	$\xrightarrow{t}$	
	$\xleftarrow{c}$	$c \in_R \mathcal{C} \subseteq \mathbb{Z}_q$
$r = k + xc$	$\xrightarrow{r}$	$h^r \stackrel{?}{=} t \cdot z^c$