Random stuff

Martin Stanek

Department of Computer Science
Comenius University
stanek@dcs.fmph.uniba.sk

Cryptology 1 (2020/21)

Content

Secret sharing schemes – introduction

- secret sharing schemes
  - distribute a secret (e.g. key) among some group of participants (users, servers)
  - rules – what group can reconstruct the secret
  - share – secret piece of information owned by individual participant
- a scheme consists of two algorithms/protocols:
  - producing and distributing the shares (usually uses a dealer)
  - reconstructing the shared secret
- motivation
  - Can you trust a single authority (admin or server)?
  - basis for other constructions – threshold cryptography, distributing computation among group of trusted servers, multi-party secure computation, electronic voting, ...

Secret sharing schemes

- n participants $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$
- shared secret s
- shares: $P_i \leftarrow s_i$
- access structure $\mathcal{A} \subseteq 2^{\mathcal{P}}$ (power set)
  - $A \subseteq \mathcal{P}$ can reconstruct $s \Leftrightarrow A \in \mathcal{A}$
  - usually monotone access structure:

$$\forall A, B \subseteq \mathcal{P} : \; A \subseteq B \;\&\; A \in \mathcal{A} \;\Rightarrow\; B \in \mathcal{A}$$

  - (t, n) threshold access structure, for $1 \leq t \leq n$:

$$\{A \mid A \subseteq \mathcal{P} \;\&\; |A| \geq t\}$$

Simple examples

- $(1, n)$ threshold
  - distribute the secret as individual shares: $s_i = s$
- (n, n) threshold – 1st attempt
  - let $s \in \{0,1\}^l$
  - divide s into n shares $s_1, \ldots, s_n$ of length $\sim l/n$ bits
  - reconstruction: $s = s_1 \,||\, \ldots \,||\, s_n$
  - $n-1$ participants reconstruct a large part of s, approx. $l(n-1)/n$ bits
- (n, n) threshold
  - let $s \in \{0,1\}^l$
  - let $s_i \xleftarrow{\$} \{0,1\}^l$ for $i = 1, \ldots, n-1$, and $s_n = s \oplus s_1 \oplus \ldots \oplus s_{n-1}$
  - reconstruction: $s = s_1 \oplus \ldots \oplus s_n$
  - security: any $n-1$ (or less) participants learn nothing about s
  - *perfect* scheme

Simple examples

- $(1, n)$ threshold
  - distribute the secret as individual shares: $s_i = s$
- (n, n) threshold – 1st attempt
  - let $s \in \{0,1\}^l$
  - divide s into n shares $s_1, \ldots, s_n$ of length $\sim l/n$ bits
  - reconstruction: $s = s_1 \,||\, \ldots \,||\, s_n$
  - $n-1$ participants reconstruct a large part of s, approx. $l(n-1)/n$ bits
- (n, n) threshold
  - let $s \in \{0,1\}^l$
  - let $s_i \xleftarrow{\$} \{0,1\}^l$ for $i = 1, \ldots, n-1$, and $s_n = s \oplus s_1 \oplus \ldots \oplus s_{n-1}$
  - reconstruction: $s = s_1 \oplus \ldots \oplus s_n$
  - security: any $n-1$ (or less) participants learn nothing about s
  - *perfect* scheme

Shamir's secret sharing scheme

- idea: t points uniquely determine some polynomial of degree $t-1$
- finite field $\mathbb{Z}_p$, for a prime $p > n$
- shared secret $s \in \mathbb{Z}_p$; let us assume $s \xleftarrow{\$} \mathbb{Z}_p$
- computing the shares:
  - choose a random polynomial $f(x) = s + a_1 x + \ldots + a_{t-1} x^{t-1}$, where $a_i \xleftarrow{\$} \mathbb{Z}_p$ for $i = 1, \ldots, t-1$
  - notice that $f(0) = s$
  - share for P_i: (i, s_i), where $s_i = f(i)$
- reconstruction; WLOG let us assume t participants $P_1, \ldots, P_t$:
  - Lagrange interpolation using (i, s_i) for $i = 1, \ldots, t$:

$$f(x) = \sum_{i=1}^{t} \underbrace{f(i)}_{s_i} \prod_{\substack{1 \le j \le t \\ j \neq i}} \frac{x-j}{i-j}$$

  - compute $s = f(0)$ (all computations are in the finite field)

Shamir's secret sharing scheme

- idea: t points uniquely determine some polynomial of degree $t-1$
- finite field $\mathbb{Z}_p$, for a prime $p > n$
- shared secret $s \in \mathbb{Z}_p$; let us assume $s \xleftarrow{\$} \mathbb{Z}_p$
- computing the shares:
  - choose a random polynomial $f(x) = s + a_1 x + \ldots + a_{t-1} x^{t-1}$, where $a_i \xleftarrow{\$} \mathbb{Z}_p$ for $i = 1, \ldots, t-1$
  - notice that $f(0) = s$
  - share for P_i: (i, s_i), where $s_i = f(i)$
- reconstruction; WLOG let us assume t participants $P_1, \ldots, P_t$:
  - Lagrange interpolation using (i, s_i) for $i = 1, \ldots, t$:

$$f(x) = \sum_{i=1}^{t} \underbrace{f(i)}_{s_i} \prod_{\substack{1 \le j \le t \\ j \ne i}} \frac{x-j}{i-j}$$

  - compute $s = f(0)$ (all computations are in the finite field)

Shamir's secret sharing scheme – security

- consider group of $t-1$ participants (WLOG $P_1, \ldots, P_{t-1}$)
- the shared secret can be anything:
  - combine the shares and add point $(0, s')$ for an arbitrary $s' \in \mathbb{Z}_p$
  - t points $\Rightarrow$ unique polynomial f'
  - f' is consistent with shares of $P_1, \ldots, P_{t-1}$
- $P_1, \ldots, P_{t-1}$ are in the same position as someone without any share
  - probability of finding $s \sim$ is $1/p$ (guessing)
- perfect secret sharing scheme

Linear equations perspective

- unknown polynomial f (its coefficients)
- a share (i, s_i) forms a linear equation: $s_i = a_0 + a_1 i + ... + a_{t-1} i^{t-1}$
- t cooperating participants – the system of t equations with t variables
  - square Vandermonde matrix with distinct elements (i.e. non-zero determinant)
  - the system has a unique solution
- $t - 1$ cooperating participants – the system of $t - 1$ equations with t variables
  - add an additional equation: $s' = a_0$
  - square Vandermonde matrix with distinct elements (because any $i \neq 0$)
  - the system has a unique solution for any s' ... perfect scheme

Remarks

- reconstruction is just a linear combination of shares:

$$f(0) = \sum_{i \in S} s_i \cdot r_i$$

  for coefficients $r_i = \prod_{j \in S \setminus \{i\}} -j/(i-j)$, and $S \subseteq \{1, \ldots, n\}$, $|S| = t$
- any points $(x_i, f(x_i))$ for distinct non-zero $x_1, \ldots, x_n$ can be used as shares
- homomorphic property with respect to addition:
  - two (t, n) threshold schemes defined by polynomials f and g
  - adding shares: $(i, f(i)), (i, g(i)) \mapsto (i, f(i) + g(i))$
  - polynomial (the shared secret is the addition of shared secrets $a_0 + a'_0$):

$$f(x) + g(x) = \sum_{i=1}^{t-1} a_i x^i + \sum_{i=1}^{t-1} a'_i x^i = \sum_{i=1}^{t-1} (a_i + a'_i) x^i$$

Remarks (2)

- efficiency
  - polynomial time
  - long s can be divided into shorter pieces and shared by independent schemes (or we can encrypt s and share the encryption key)
- trusted dealer – generates the polynomial and distributes the shares
- one-time scheme?
  - secret revealed after reconstruction vs. black-box reconstruction
- cheating in reconstruction:
  - for example – $P_1, \ldots, P_t$ try to reconstruct s
  - P_1 cheats and reveals an incorrect share $(1, s_1')$
  - the participants compute: $s' = s + s_1' r_1 - s_1 r_1$
    …and P_1 can easily compute s from s'

Information rate

- the size of share(s) vs. the size of the shared secret
- notation
  - S – set of secrets
  - $K(P_i)$ – set of all possible shares for P_i
  - random variables
- information rate for P_i: $\rho_i = H(S)/H(K(P_i))$
- information rate of the scheme: $\rho = \min_i \rho_i$
- uniform probability case: $\rho = \min_i \lg|S|/\lg|K(P_i)|$

Information rate (2)

- information rate for Shamir's scheme: $\rho = 1$
- perfect secret sharing scheme ... $\rho \leq 1$
  - let us assume that $\rho > 1 \implies \forall i : \rho_i > 1$
  - for all i:

$$\begin{aligned} \lg |S| / \lg |K(P_i)| &> 1 \\ \lg |S| &> \lg |K(P_i)| \\ |S| &> |K(P_i)| \end{aligned}$$

  - there exists $A \subseteq \mathcal{P}$: $P_i \notin A$, $A \notin \mathcal{A}$, and $A \cup \{P_i\} \in \mathcal{A}$
  - take all shares from participants in A and all candidate shares from $K(P_i)$
  - compute all possible values of the shared secret ... less than $|S|$
  - the scheme cannot be perfect (we can exclude some "impossible" secrets)
- a perfect secret sharing scheme with $\rho = 1$ is called ideal

Commitment schemes – introduction

- How to flip a coin or play rock-paper-scissors(-lizard-Spock) over phone/network?
  - "no, it was a head/tail" (or whatever suits me)
  - we want to guarantee a fairness
- commitment scheme: participant "commits" himself to some value
  - later, the participant opens the commitment to show the value
  - the commitment does not allow to compute the value (efficiently)
  - the participant cannot open the commitment as a different value
- bit commitment
- flipping a coin (two participants):
  1. A commits to a bit b
  2. B guesses b'
  3. A opens the commitment; result can be defined as $b \oplus b'$

Commitment schemes – properties

- commitment scheme: two-phase protocol between Sender and Receiver
  1. commit: Sender produces a commitment of some value m for Receiver
  2. reveal: Sender opens the commitment for Receiver
     Receiver can verify the correctness of the commitment
- hiding
  - Receiver cannot compute anything about m from its commitment
  - computational vs. perfect hiding
- binding
  - Sender cannot open the commitment as a different value $m' \neq m$
  - computational vs. perfect binding

Examples

- analogy: a box with a lock
  - commit: Sender writes a value on a paper and locks it inside the box. Sender gives the box to Receiver.
  - reveal: Sender gives the key to Receiver.
  - properties:
    binding (Sender does not have the box),
    hiding (Receiver does not have the key)
- Ad-hoc construction from hash function: $f(b, r) = H(b \,||\, r)$
  - $b \in \{0, 1\}$, random r
  - reveal: show b and r
  - binding $\sim$ collision resistance
  - hiding $\sim$ inability to find the first bit of preimage (this is weaker than preimage resistance)

Examples 2

- RSA based construction: $r^e \bmod n$ (where $r \bmod 2 = b$)
  - random $r \in \mathbb{Z}_n$
  - reveal: show r
  - perfect binding (unconditionally), RSA encryption is a permutation on $\mathbb{Z}_n$
  - computational hiding: RSA asumption, security of plaintext parity bit

- Pedersen bit-commitment scheme
  - $(G, \cdot)$ cyclic group of prime order q
  - g, h – generators with unknown $\mathrm{dlog}_g h = x$
  - commit: $f(b, r) = g^r h^b$ for random $r \in \mathbb{Z}_q$
  - reveal: show b and r
  - computational binding: find r, r' such that $g^r h = g^{r'} \implies h = g^{r'-r}$ yields x
  - perfect hiding: any commitment $c \in G$ can be
    0: there exists r such that $g^r = c$
    1: there exists r such that $g^r = ch^{-1}$
  - the scheme can be used for arbitrary values $b \in \mathbb{Z}_q$

Perfectly hiding & perfectly binding scheme

- you can have one or other but you cannot have both properties
- let us discuss a *bit*-commitment scheme
- assume the existence of such scheme
- for any commitment c of 0 there exists r': $f(r', 1) = c$ (because of perfect hiding); therefore it cannot be perfectly binding (unlimited Sender can find r')

Interactive proof systems (IPS)

- What is a proof?
- communicating parties: P – prover, V – verifier
- protocol, modeled as a pair of interactive Turing machines
  - common input x
  - V is probabilistic polynomial time
  - P is computationally unlimited
  - V accepts or rejects at the end (we say that (P, V) accepts/rejects)
- let $L \subseteq \{0,1\}^*$ be some language
- P tries to "convince" V that $x \in L$

IPS – definition

- (P, V) is an IPS for L if
  1. completeness: $\forall x \in L : \Pr[(P, V)(x) \text{ accepts}] \geq 2/3$
  2. soundness: $\forall x \notin L \, \forall P^* : \Pr[(P^*, V)(x) \text{ accepts}] \leq 1/3$

- reducing error probabilities by sequential iteration of IPS
  - taking a majority of accept/reject votes
  - apply Chernoff's bound
    - let $X_1, \ldots, X_k$ be independent 0/1 random variables, with $\Pr[X_i] = p$ for all i and some probability p; then for $0 < \delta < 1$:

$$\Pr\left[\sum_i X_i \leq (1-\delta)pk\right] \leq e^{-\frac{\delta^2 pk}{2}}$$

  - n independent repetitions
    - error probability at most $2^{-f(n)}$, for some polynomial f

IP class

- IP – class of languages with IPS
- IP = PSPACE
- large class of languages
- QSAT is a complete language for PSPACE:

$$(Q_1\, x_1)(Q_2\, x_2) \ldots (Q_n\, x_n)\ \varphi(x_1, x_2, \ldots, x_n)$$

IPS – remarks

- V is polynomial $\Rightarrow$
  - polynomial number of rounds
  - polynomial length of messages
- perfect completeness & soundness
  - perfect completeness does not change the power of IPS
  - perfect soundness results in NP
- "ingredients" of IPS
  - randomness – without randomness (deterministic verifier) we get NP
  - interaction – without interaction
    single message from P to V ... MA (Merlin-Arthur) class
    no message at all ... BPP class
  - private coins not necessary – public coins do not change the power of IPS

Graph non-isomorphism

- GNI $= \{(G_1, G_2) \mid G_1 \not\simeq G_2\}$
- GNI seems to be outside of NP (trivially GI $\in$ NP)
- IPS for GNI:
  1. $V \to P$: H, random isomorphic copy $H \simeq G_i$ for $i \xleftarrow{\$} \{1, 2\}$
  2. $P \to V$: i'
  3. if $i \neq i'$ then V rejects

  V accepts after k successful iterations

Graph non-isomorphism 2

- completeness: let $G_1 \not\simeq G_2$ (i.e. $(G_1, G_2) \in \mathrm{GNI}$)
  - H is isomorphic to just one G_i
  - P can always succeed, i.e. $\Pr[(P, V)(G_1, G_2) \text{ accepts}] = 1$
- soundness: let $G_1 \simeq G_2$ (i.e. $(G_1, G_2) \notin \mathrm{GNI}$)
  - H is isomorphic to both G_1, G_2
  - any P^* can only guess the correct value of i
  - probability of success in a single round is $1/2$
  - $\Pr[(P^*, V)(G_1, G_2) \text{ accepts}] \leq 2^{-k}$

Remarks

- completeness assumes an honest prover (and verifier)
  - completeness is not about security
  - sometimes it can be easy to prove that $x \in L$ if it is true
- soundness protects the verifier against accepting $x \notin L$ while interacting with malicious prover
- consider $\mathrm{GI} = \{(G_1, G_2) \mid G_1 \simeq G_2\}$
  - easy to design an IPS for GI
  - P can compute and send the isomorphism $\varphi : G_1 \to G_2$ to V
  - V can verify φ in polynomial time
  - completeness & soundness 100%
- similar idea can be used for any $L \in \mathrm{NP}$ (e.g. SAT, HAM, ...)

Graph isomorphism

- another IPS for GI:
  1. $P \to V$: H, random isomorphic copy $H \simeq G_1$
  2. $V \to P$: $i \xleftarrow{\$} \{1, 2\}$
  3. $P \to V$: π (permutation on vertices of G_i)
  4. V checks if π is isomorphism of G_i and H; if not then V rejects

  V accepts after k successful iterations
- completeness: let $G_1 \simeq G_2$ (i.e. $(G_1, G_2) \in \mathrm{GI}$)
  - P can compute isomorphism between H and any G_i
  - P can always answer with correct π
- soundness: let $G_1 \not\simeq G_2$ (i.e. $(G_1, G_2) \notin \mathrm{GI}$)
  - P^* sends some graph H in step 1
  - H is isomorphic to at most one graph G_i
  - P^* succeeds in single round with probability $\leq 1/2$
  - V accepts with probability at most 2^{-k}

Zero-knowledge IPS with honest verifier

- P wants to prove $x \in L$ without providing anything beyond this fact
- zero-knowledge
  - V will not learn anything that he cannot compute by himself
- $\text{view}(P, V, x)$ – random variable containing transcript of $P \leftrightarrow V$ communication on input x
- idea: the communication does not contain any knowledge if it can be simulated efficiently
- IPS (P, V) for L is perfect zero-knowledge for honest verifier if

$$\exists \text{ PPT } S \; \forall x \in L : \; \text{view}(P, V, x) = S(x)$$

- the distributions of real and simulated communications are equal

Simulating IPS for GI

- S works as follows (repeating k times):
  1. choose $i \xleftarrow{\$} \{1, 2\}$
  2. choose random permutation π
  3. output: $(\pi(G_i), i, \pi)$
- each triple is distributed identically (we assume $G_1 \simeq G_2$) to original $P \leftrightarrow V$ communication

Perfect ZK IPS

- verifier can be malicious
  - V^* tries to get as much knowledge as possible from the prover
  - V^* can deviate from the protocol (selecting his challenge according some "strategy")

- IPS (P, V) for L is perfect zero-knowledge if

$$\forall \text{PPT } V^* \; \exists \text{ PPT } S \; \forall x \in L : \; \text{view}(P, V^*, x) = S(x)$$

Perfect ZK IPS for GI

- the IPS for GI is perfect zero-knowledge
  - black-box simulation, S remembers the state of V^*
  - single round:

1. S chooses random i' and π'
2. S computes $H' = \pi'(G_{i'})$
3. S simulates V^* on message H'
4. S obtains a challenge i
5. if $i \neq i'$ then S resets V^* into previous state and the round starts again
6. output: (H', i', π'); S remembers new state of V^*

- producing H' and π' is exactly how P works
- the choice of i is the genuine V^*'s choice
- i' is independent on i, i.e. $\Rightarrow \Pr[i = i'] = 1/2$
- expected number of repetitions: $1 + 1/2 + 1/4 + \ldots = 2$, i.e. S runs in PPT

Remarks

- types of ZK (wrt simulated and real communications)
  - perfect – equal/identical
  - statistical – negligible statistical difference
  - computational (CZK) – indistinguishable in PPT, i.e. the probability of distinguishing the distributions is negligible
- CZK = IP (if one-way functions exist)
- let show NP $\subseteq$ CZK
  - HAM = $\{G \mid G$ has an Hamiltonian cycle$\}$
  - NP-complete language

CZK IPS for HAM

- repeat k times for an input G (V accepts after k successful rounds):
  1. $P \to V : c(H)$, where H is a random isomorphic copy of G, and $c(H)$ is a commitment of H (commitments of all values in its incidence matrix)
  2. $V \to P : c \xleftarrow{\$} \{0, 1\}$
  3. $P \to V$:
     if $c = 0$: P opens all commitments and sends π: $\pi(G) = H$
     if $c = 1$: P opens those commitments that reveal a Hamiltonian cycle in $c(H)$
  4. V verifies received data and rejects if there is anything wrong
- completeness: if $G \in$ HAM then P always succeeds
- soundness: $G \notin$ HAM
  - assume perfectly binding BC scheme (because P^* is unlimited)
  - P^* succeeds in single round $\Leftrightarrow$ correct guess of the challenge in advance
  - probability of success after k rounds $\leq 2^{-k}$

CZK IPS for HAM

- repeat k times for an input G (V accepts after k successful rounds):
  1. $P \to V : c(H)$, where H is a random isomorphic copy of G, and $c(H)$ is a commitment of H (commitments of all values in its incidence matrix)
  2. $V \to P : c \xleftarrow{\$} \{0, 1\}$
  3. $P \to V$:
     if $c = 0$: P opens all commitments and sends π: $\pi(G) = H$
     if $c = 1$: P opens those commitments that reveal a Hamiltonian cycle in $c(H)$
  4. V verifies received data and rejects if there is anything wrong
- completeness: if $G \in$ HAM then P always succeeds
- soundness: $G \notin$ HAM
  - assume perfectly binding BC scheme (because P^* is unlimited)
  - P^* succeeds in single round $\Leftrightarrow$ correct guess of the challenge in advance
  - probability of success after k rounds $\leq 2^{-k}$

CZK IPS for HAM 2 – zero-knowledge

- black-box simulation for single round:
  1. S chooses $c' \xleftarrow{\$} \{0, 1\}$
  2. if $c' = 0$ then S selects random π and computes $c(\pi(G))$
  3. if $c' = 1$ then S chooses Hamiltonian graph H and computes $c(H)$
  4. S simulates V^* and obtains c
  5. if $c \neq c'$ then reset V^* into previous state and start again
  6. otherwise S can produce the output; remember a new state of V^*
- simulated communication is not identical to the real one
  - computationally unbounded distinguisher can distinguish them
  - they cannot be distinguished in PPT (BC is computationally hiding)
  - CZK
- we need efficient provers for practical applications ...