

Share conversion, pseudorandom secret-sh
applications to secure distributed comp

Ronald Cramer (CWI & Leiden University), *Ivan*
(Aarhus University), *Yuval Ishai* (Technio

Friday, February 11, 2005

(t, n) Replicated secret sharing

- Consider all $\binom{n}{t}$ subsets $B \subset \{1, \dots, n\}$ with $|B| = t$; these subsets *cells*.
- Additively secret-share the secret s , where each share s_B is replicated among the t players in the i -th cell B_i .

In other words the i -th share is replicated among the t players in the i -th cell B_i .

- Thus: $s = \sum_B s_B$, and player P_j holds $\{s_B\}_{B:j \in B}$.

Privacy:

- Consider $A \subset \{1, \dots, n\}$ with $|A| = t$.

There is a cell B that has empty intersection with

$$B = \{1, \dots, n\} \setminus A$$

- So, A lack the share s_B .

Reconstruction:

- All players jointly can determine the secret s .
- With $n > 2t$: the intersection of any two cells is
- So each B jointly have all shares, and can can r

[Example: if $n = 2t + 1$, then t -private, $> t$ reco

- $n > 3t \Rightarrow$ perfect recovery from malicious error construction:
 - In each cell, there is a majority of good guys ($n > 2t$)
 - So find correct s_B by *local* “majority voting” shares received from members of B

Drawback: efficiency proportional to $\binom{n}{t}$.

Why interested in this scheme...???

After all...there is Shamir's scheme..

Ito/Nishizeki/Saito: introduced general, non-thresholding secret sharing (m-out-of m additive sharing within a "qualified set")

Beaver/Wool: simple protocol for MPC, passive adversarial adversary

Maurer: extension to active case

But: only makes sense when n is small (or more generally t is small).

Moreover, in all these cases more efficient solutions are found by more sophisticated techniques

Good reasons to revisit this technique to follow...

Pseudo-random secret sharing

Unbounded source of sharings of random secrets, no

- Trusted Initialization: replace share s_B by a pseudo-random function $G_B(\cdot)$.

- Define

$$s(\cdot) = \sum_B G_B(\cdot).$$

Notation: suppress globally agreed argument (\cdot)

- Variation: seeds for pseudo-random number generation

Pseudo-random secret sharing introduced earlier by M
and further studied by *Ishai/Gilboa*,...

Here: we develop enhanced pseudo-random secret sh
and give new applications of pseudo-random secret

*taking interaction out of certain secure computation
crypto-systems, without paying a penalty in commu*

Application I: Pseudo-random VSS with $t <$

- Give all pseudo-random functions to a single designer (the dealer): *non-interactive VSS* of random secret s by dealer.
- *Adaptation to secret s of his choice*: dealer broadcasts reconstruction value $s - r$.
- Players adapt their shares *locally* using linearity:
- *Reconstruction* in presence of malicious errors: error correction

Application II: Non-interactive secure multiplication

Here: $n > 4t$. Adversary actively corrupts at most t

- Intersection of any two cells contains majority of

For each pair of cells B, B' , designate a (unique)
 $B \cap B'$ of size $2t + 1$. Call this a subcell.

- *Initialization*: for each subcell, replicate a fresh
the pseudo-random VSS set-up.

- *Starting condition:* sharings of α, β in the replication sharing scheme. Thus:

$$\alpha = \sum_B \alpha_B, \quad \beta = \sum_B \beta_B, \quad \alpha \cdot \beta = \sum_{B, B'} \alpha_B \cdot \beta_{B'}$$

- *Basic Idea:*

For each subcell C : usual re-sharing of local product but done with pseudo-random VSS instead

Re-sharings of a correct local products occur in

This is due to pseudo-random VSS replicas; re-voting over the broadcasted correction value

Bunch it up non-interactively to a sharing of the

Note 1: No broadcast needed here in pseudo-random

Note 2: (information theoretic pre-processing versus random approach)

preprocessing leads to a priori bounded horizon, pseudorandomness makes it ongoing, unbounded horizon

Compressed pseudo-random secret sharing

New feature: *share size* “as in Shamir”, still non-int

Only local *computation* proportional to $\binom{n}{t}$

- For each cell B , choose fixed polynomial f_B so
 1. $\deg(f) = t$
 2. $f(0) = 1$ but $f(i) = 0$ if $i \notin B$
- Define

$$f(X) = \sum_B s_B \cdot f_B(X)$$

Player P_i can compute his Shamir-share from his share replication based scheme:

$$\begin{aligned} f(i) &= \\ \sum_B s_B \cdot f_B(i) &= \\ \sum_{B:i \in B} s_B \cdot f_B(i) + \sum_{B:i \notin B} s_B \cdot f_B(i) &= \\ &= \sum_{B:i \in B} s_B \cdot f_B(i) \end{aligned}$$

Privacy: the info held by $A \subset \{1, \dots, n\}$ with $|A| \geq t$ is the value of $f(X)$.
least one random coin in the expression for $f(X)$...

Compressed pseudo-random VSS

This works as before (non-interactive), with $n > 3t$

All pseudo-random functions given to designated pl

Broadcast difference of pseudo-random secret and
terest

Reconstruction: efficient Reed-Solomon decoding (s
Welch)

Compressed non-interactive secure multiplication

$$n > 4t$$

Adversary actively corrupts t players

- *Initialization*: compressed pseudo-random secret share place (actually, a small variation...)

- *Input*: (t, n) -Shamir sharings

$$(\alpha_1, \dots, \alpha_n), (\beta_1, \dots, \beta_n)$$

of secrets α, β

- *Output*: $\alpha \cdot \beta$.

Pseudo-random zero-sharing: (with $n > 4t$) create a Shamir-sharing of degree $2t$ of the value 0.

- For each cell B , choose a *fixed basis* of polynomials for the *vector space* of polynomials f with
 1. $\deg(f) \leq 2t$
 2. $f(0) = 0$
 3. $f(i) = 0$ if $i \notin B$

- Instead of a single one, hand pseudo-random fu

$$G_B^1(\cdot), \dots, G_B^t(\cdot)$$

to the players in B .

- Define

$$f(X) = \sum_B \sum_{i=1}^t s_B^i \cdot f_B^i(X)$$

Compressed secure multiplication is now easy, by standard technique plus masking using pseudo-random zero-sharing

- γ_i : player i 's share in the pseudo-random zero-sharing
- *Masking*: player i computes

$$\alpha_i \cdot \beta_i + \gamma_i,$$

sends it to all players

- Each player on his own applies Berlekamp-Welch correction.

- Indeed:

f : polynomial for zero-sharing

f_α, f_β : polynomials for sharing of α, β

So $f_\alpha \cdot f_\beta + f$ of degree $\leq 2t$,

we have $n > 4t$ points with $\leq t$ errors.

So the error correction is possible.

- This way, each player obtains a polynomial who is $\alpha \cdot \beta$.

Generalization: non-interactive secure computation
bi-variate polynomials

Theoretical Results on Share Compression

- Thm.: *Pseudo-random secret sharing schemes can be compressed to any linear secret sharing scheme*

Proof: generalize the Shamir compression using monotone span programs

- Thm.: *Our approach is optimal in the model where a player gets a subset of a given collection of independent distributed random sources*

Proof: By information theoretic arguments: $\# \text{ random sources} \geq \# \text{ maximal unqualified sets}$

Application to non-interactive threshold cry

- Non-interactive version of the threshold-CS98 f Goldwasser:

test of validity of ciphertext by non-interactive tionby (compressed secure multiplication “in the

- Communication-efficient variant of Naor/Pinkas/tributed Pseudo-Random Function
- Threshold signatures without random oracles bas Boyen scheme.