ASIACRYPT 2014

Compact VSS and Efficient Homomorphic UC Commitments

Ivan Damgård, Bernardo David, Irene Giacomelli and Jesper B. Nielsen

Aarhus University

Road-map:

1. **Verifiable Secret-Sharing Scheme** (VSS)

2. **Homomorphic UC Commitment Scheme**

Road-map:

1. **Verifiable Secret-Sharing Scheme** (VSS)
   - based on any Linear Secret-Sharing Scheme (LSSS);
   - compact: many secrets shared in one execution
     $\longrightarrow$ communication rate $O(1)$;
2. **Homomorphic UC Commitment Scheme**

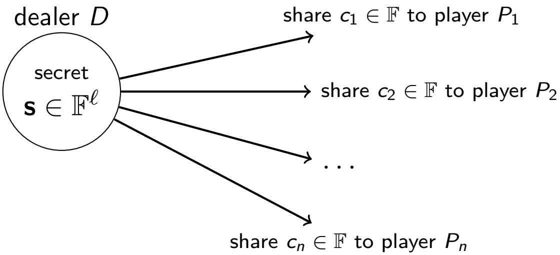
Road-map:

1. **Verifiable Secret-Sharing Scheme** (VSS)
   - based on any Linear Secret-Sharing Scheme (LSSS);
   - compact: many secrets shared in one execution
     $\longrightarrow$ communication rate $O(1)$;

2. **Homomorphic UC Commitment Scheme**
   - based on the VSS in a "MPC-in-the-head" setting [IKOS07, IPS08];
   - designed in the OT-hybrid model using preprocessing;
   - efficient: $\longrightarrow$ linear comput. complexity for the receiver.

Packed Linear Secret-Sharing Scheme among n players

Sharing Phase:



Reconstruction Phase:

$$\left.\begin{array}{l} \text{share } c_1 \in \mathbb{F} \\ \text{share } c_2 \in \mathbb{F} \\ \qquad\vdots \\ \text{share } c_n \in \mathbb{F} \end{array}\right\} \longrightarrow \mathbf{s} \in \mathbb{F}^{\ell}$$

Packed Linear Secret-Sharing Scheme among n players

Sharing Phase:

dealer D

secret $\mathbf{s} \in \mathbb{F}^\ell$

share $c_1 \in \mathbb{F}$ to player P_1

share $c_2 \in \mathbb{F}$ to player P_2

...

share $c_n \in \mathbb{F}$ to player P_n

t-privacy

Any set of at most t shares gives no info on $\mathbf{s}$

r-reconstruction

Any set of at least r shares fully determines $\mathbf{s}$

Reconstruction Phase:

$$\left.\begin{array}{l} \text{share } c_1 \in \mathbb{F} \\ \text{share } c_2 \in \mathbb{F} \\ \quad\vdots \\ \text{share } c_n \in \mathbb{F} \end{array}\right\} \longrightarrow \mathbf{s} \in \mathbb{F}^\ell$$

$$1 \leq t < r \leq n$$

$\ell =$ secret length $(\ell > 1)$

Sharing Phase in LSSS:

LSSS $\longleftrightarrow$ a $n \times (\ell + e)$ public matrix $\mathbf{M}$

Sharing Phase in LSSS:

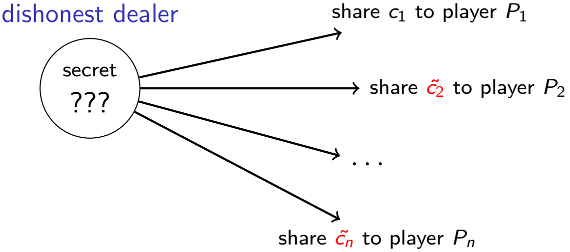
LSSS $\longleftrightarrow$ a $n \times (\ell + e)$ public matrix $\mathbf{M}$

1. D chooses $\mathbf{f} = \begin{pmatrix} | \\ \mathbf{s} \\ | \\ | \\ \mathbf{v} \\ | \end{pmatrix}$ $\begin{array}{l} \leftarrow \text{the secret, column vector in } \mathbb{F}^{\ell} \\ \\ \\ \leftarrow \text{the randomness, column vector in } \mathbb{F}^{e} \end{array}$

2. D computes $\begin{pmatrix} \mathbf{c}[1] \\ | \\ \mathbf{c}[n] \end{pmatrix} = \mathbf{M} \cdot \mathbf{f}$ and sends $\mathbf{c}[i]$ to P_i

Security: the players' point of view

What happens if the dealer is not honest?!



$$\left.\begin{array}{l}\text{share } c_{i_1} \in \mathbb{F}\\ \text{share } c_{i_2} \in \mathbb{F}\\ \qquad\vdots\\ \text{share } c_{i_r} \in \mathbb{F}\end{array}\right\} \longrightarrow \mathbf{s} \in \mathbb{F}^{\ell} \qquad \left.\begin{array}{l}\text{share } \tilde{c}_{j_1} \in \mathbb{F}\\ \text{share } \tilde{c}_{j_2} \in \mathbb{F}\\ \qquad\vdots\\ \text{share } \tilde{c}_{j_r} \in \mathbb{F}\end{array}\right\} \longrightarrow \tilde{\mathbf{s}} \in \mathbb{F}^{\ell}$$

Definition of VSS Scheme [CGMA85]

A (t, r)-LSSS among n players is **verifiable** if

- t-**privacy**: no info from t shares

$$\left.\begin{array}{c} P_{i_1} \\ P_{i_2} \\ \vdots \\ P_{i_t} \end{array}\right\} \longrightarrow ?$$

Definition of VSS Scheme [CGMA85]

A (t, r)-LSSS among n players is **verifiable** if

- t-**privacy**: no info from t shares

$$\left.\begin{array}{l} P_{i_1} \\ P_{i_2} \\ \vdots \\ P_{i_t} \end{array}\right\} \longrightarrow ?$$

- r-**robust reconstruction**: when the dealer is corrupt,

  the sharing phase succeeds

  $\Downarrow$

  any set of r honest players reconstruct <u>the same secret</u>

For any $\{i_1, \ldots, i_r\} \neq \{j_1, \ldots, j_r\}$, if

$$\left.\begin{array}{l} P_{i_1} \\ P_{i_2} \\ \vdots \\ P_{i_r} \end{array}\right\} \longrightarrow \mathbf{s} \in \mathbb{F}^\ell \quad \text{and} \quad \left.\begin{array}{l} P_{j_1} \\ P_{j_2} \\ \vdots \\ P_{j_r} \end{array}\right\} \longrightarrow \tilde{\mathbf{s}} \in \mathbb{F}^\ell$$

$$\Longrightarrow \mathbf{s} = \tilde{\mathbf{s}}$$

Known constructions from LSSS to VSS

- in [BGW88] $\rightarrow$ verifiable version of Shamir's LSSS
  (only secrets of length 1!)

- in [FY92] $\rightarrow$ packed version of Shamir's LSSS (no verifiable!)

- in [CDM00]$\rightarrow$ generalization of the previous schemes:
  - it works for more general LSSS;
  - only secrets of length 1;
  - it has communication complexity $O(n)$
    (n is the number of the players).

- our construction $\rightarrow$ verifiable, works for general LSSS, secrets of any length

Our construction from LSSS to VSS:

(t, r)-LSSS for secret $\mathbf{s} \in \mathbb{F}^{\ell}$ $\Rightarrow$ (t, r)-VSS for secrets $\{\mathbf{s}_1, \ldots, \mathbf{s}_\ell\} \subseteq \mathbb{F}^{\ell}$

	field elements shared	communication complexity
LSSS	ℓ	$\Theta(n)$
VSSS	ℓ^2	$\Theta(n^2)$

Assuming $\ell = \Theta(n)$, **constant rate!**

Sharing Phase in our VSS

Secrets $\{\mathbf{s}_1, \dots, \mathbf{s}_\ell\} \subseteq \mathbb{F}^\ell$, LSSS-matrix $\mathbf{M}$ with rows $\mathbf{m}_i$

- D chooses $\mathbf{F} = \begin{array}{c} \text{secrets} \\ \left(\begin{array}{cccccc} | & \dots & | & * & & * \\ \mathbf{s}_1 & \dots & \mathbf{s}_\ell & \vdots & & \vdots \\ | & \dots & | & * & & * \\ * & \dots & * & * & \dots & * \\ \vdots & & \vdots & \vdots & & \vdots \\ * & \dots & * & * & \dots & * \end{array}\right) \end{array} \begin{array}{l} \\ \leftarrow \text{randomness} \\ \\ \\ \leftarrow \text{randomness} \\ \\ \end{array}$

- D computes $\mathbf{g}_i = \mathbf{m}_i \cdot \mathbf{F}$ (row vector) and $\mathbf{h}^i = \mathbf{F} \cdot \mathbf{m}_i^\top$ (column vector)

$$\underset{\substack{\uparrow \\ \textit{public}}}{\mathbf{m}_i} \cdot \underset{\substack{\uparrow \\ P_j}}{\mathbf{h}^j} = \underset{\substack{\uparrow \\ P_i}}{\mathbf{g}_i} \cdot \underset{\substack{\uparrow \\ \textit{public}}}{\mathbf{m}_j^\top}$$

Our construction from LSSS to VSS: extensions

- (strong) multiplication property inherited from the LSSS;
- checking a public linear relation between the secrets;

  D shares $\mathbf{s}$ and $\mathbf{s}'$, the players can check if $\varphi(\mathbf{s}) = \mathbf{s}'$

  (φ public linear map)
- generate shares of $\mathbf{0} \in \mathbb{F}^\ell$

Our construction from LSSS to VSS: applications

Given an underlying LSSS [CCCX09] with

t-strong multiplication

$|\mathbb{F}|$ constant

$t, \ell = \Theta(n)$

VSS $\Longrightarrow$

- MPC protocol for a circuit C over any field
  - UC perfectly secure in the client/server model;
  - C is well-formed $\rightarrow$ comm. compl. $O(|C| log |C|)$;
  - C is regular $\rightarrow$ comm. compl. $O(|C|)$.

[DIKNS08, DIK10] $\rightarrow$ similar result but with $|\mathbb{F}| \geq n$

Our construction from LSSS to VSS: applications

Given an underlying LSSS [CCCX09] with

t-strong multiplication

$|\mathbb{F}|$ constant

$t, \ell = \Theta(n)$

VSS $\Longrightarrow$

- MPC protocol for a circuit C over any field
  - UC perfectly secure in the client/server model;
  - C is well-formed $\rightarrow$ comm. compl. $O(|C| log |C|)$;
  - C is regular $\rightarrow$ comm. compl. $O(|C|)$.

  [DIKNS08, DIK10] $\rightarrow$ similar result but with $|\mathbb{F}| \geq n$

- UC Commitment Scheme!

Commitment Scheme:

Sender | **Receiver**

Commit Phase:

put the secret $\mathbf{s} \in \mathbb{F}^{\ell}$ in a locked box

$\longrightarrow$ store the box

Open Phase:

$\longrightarrow$ open the box

Commitment Scheme:

	Sender	Receiver
Commit Phase:	put the secret $\mathbf{s} \in \mathbb{F}^\ell$ in a locked box	
	 $\longrightarrow$	store the box
Open Phase:	 $\longrightarrow$	open the box

- **Hiding property**: a corrupted receiver has no info on the secret contained in a locked box sent by an honest sender

Commitment Scheme:

Sender **Receiver**

Commit Phase:

put the secret $\mathbf{s} \in \mathbb{F}^{\ell}$ in a locked box

store the box

Open Phase:

open the box

- **Binding property**: a corrupted sender can not change secret, after having sent the box to an honest receiver

Previous Commitment Schemes:

stand-alone model:
one-way function $\Rightarrow$ commitment schemes
PRG $\Rightarrow$ very efficient commitment scheme [Nao91]

UC model:
UC commitments need set-up assumptions [CF01]. Up to this year:

- Most efficient UC commitments [Lin11,BCPV13] requires exponentiations in DDH groups. $\Omega(\ell^3)$ comp. complexity.

Independent work in Eurocrypt 2014 [GIKW14]:

- optimal communication rate
- public-key crypto only in the setup phase
- relies specifically on [FY92] (packed Shamir's LSSS)
- no homomorphic properties

Our Commitment Scheme:

- public-key crypto only in the setup phase
- additively homomorphic and check multiplicative relations between commitments
- based on general LSSS
- Amortized complexity: to commit to a message of length ℓ

	Sender	Receiver	Comm. Compl.
Shamir LSSS	$O(\ell \cdot polylog(\ell))$	$O(\ell \cdot polylog(\ell))$	$O(\ell \cdot polylog(\ell))$
AG LSSS	$O(\ell^{1+\epsilon})$	$O(\ell)$	$O(\ell)$

assuming efficient PRG [VZ12]

Our Commitment Scheme, the idea:

Commit Phase on input $\{\mathbf{s}_1, \ldots, \mathbf{s}_\ell\} \subseteq \mathbb{F}^\ell$:

(Step 1)

Sender **Receiver**

(r,t)-**VSS** on $\mathbf{s}_1, \ldots, \mathbf{s}_\ell$

$$\longrightarrow \mathbf{C} = \begin{pmatrix} | & \ldots & | \\ \mathbf{c}_1 & \ldots & \mathbf{c}_\ell \\ | & \ldots & | \end{pmatrix}$$

$\mathbf{c}_i \longrightarrow$ *verifiable* share vector for the secret $\mathbf{s}_i$

row j of $\mathbf{C} \longrightarrow$ view of P_j in the VSS scheme

Our Commitment Scheme, the idea:

Commit Phase on input $\{\mathbf{s}_1, \ldots, \mathbf{s}_\ell\} \subseteq \mathbb{F}^\ell$:

(Step 1)

$\boxed{\textbf{Sender}}$

(r, t)-**VSS** on $\mathbf{s}_1, \ldots, \mathbf{s}_\ell$

$$\longrightarrow \mathbf{C} = \begin{pmatrix} | & \ldots & | \\ \mathbf{c}_1 & \ldots & \mathbf{c}_\ell \\ | & \ldots & | \end{pmatrix}$$

$\boxed{\textbf{Receiver}}$

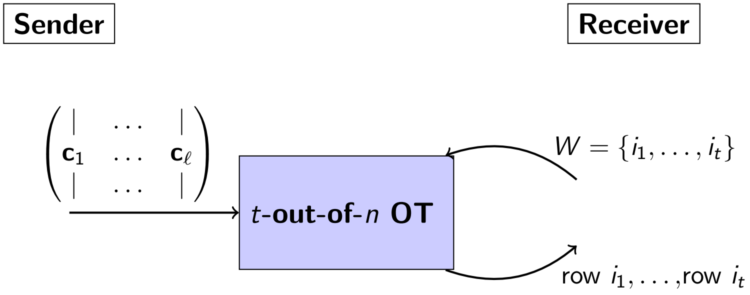
choose a random
$W = \{i_1, \ldots, i_t\}$
(watchlist)

$\mathbf{c}_i \longrightarrow$ *verifiable* share vector for the secret $\mathbf{s}_i$

row j of $\mathbf{C} \longrightarrow$ view of P_j in the VSS scheme

Our Commitment Scheme, the idea:

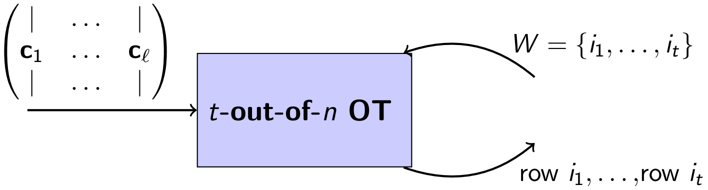
Commit Phase (Step 2)



Our Commitment Scheme, the idea:

Commit Phase (Step 2)

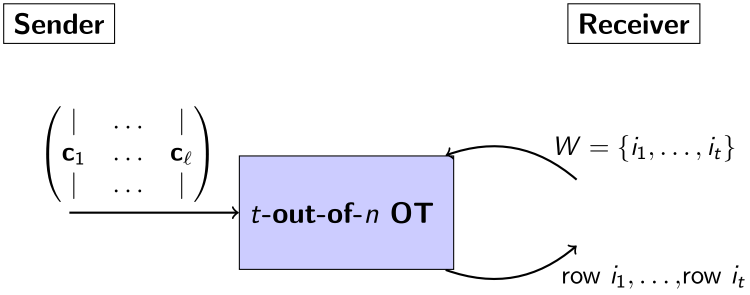
Sender **Receiver**



check the t shares
as the players in
the VSS scheme

Our Commitment Scheme, the idea:

Commit Phase (Step 2)



Hiding property

the receiver sees only t shares $\rightarrow$ no info on the secrets $\mathbf{s}_i$

(t-privacy)

check the t shares as the players in the VSS scheme

Our Commitment Scheme, the idea:

Open Phase for the secret $\mathbf{s}_i$

$$\boxed{\textbf{Sender}} \qquad\qquad\qquad\qquad \boxed{\textbf{Receiver}}$$

$$\xrightarrow{\quad\quad\quad\quad \mathbf{c}'_i \quad\quad\quad\quad}$$

check

$\mathbf{c}'_i[j] = \mathbf{c}_i[j]$

for $j \in W$

accept/reject

$W \longrightarrow$ watchlist from the Commit Phase

$\mathbf{c}_i \longrightarrow$ share vector from the Commit Phase

Our Commitment Scheme, the idea:

Open Phase for the secret $\mathbf{s}_i$

Sender $\xrightarrow{\mathbf{c}'_i}$ **Receiver**

check
$\mathbf{c}'_i[j] = \mathbf{c}_i[j]$
for $j \in W$

$$\mathbf{c_i} = \begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}$$

Binding property

- VSS checks $\Rightarrow$ in $\mathbf{c}_i$ there are $n - \epsilon$ consistent shares

Our Commitment Scheme, the idea:

Open Phase for the secret $\mathbf{s}_i$

$$\boxed{\textbf{Sender}} \qquad\qquad\qquad\qquad \boxed{\textbf{Receiver}}$$

$$\xrightarrow{\mathbf{c}_i'}$$

check
$\mathbf{c}_i'[j] = \mathbf{c}_i[j]$
for $j \in W$

$$\mathbf{c_i}' = \begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}$$

Binding property

- VSS checks $\Rightarrow$ in $\mathbf{c}_i$ there are $n - \epsilon$ consistent shares
- r-reconstruction $\Rightarrow$ S has to change $\geq n(1-\epsilon) - r + 1$ shares

Our Commitment Scheme, the idea:

Open Phase for the secret $\mathbf{s}_i$

Sender $\xrightarrow{\mathbf{c}'_i}$ **Receiver**

check
$\mathbf{c}'_i[j] = \mathbf{c}_i[j]$
for $j \in W$

$$\mathbf{c_i}' = \begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}$$

Binding property

- VSS checks $\Rightarrow$ in $\mathbf{c}_i$ there are $n - \epsilon$ consistent shares
- r-reconstruction $\Rightarrow$ S has to change $\geq n(1-\epsilon) - r + 1$ shares
- S doesn't know W $\Rightarrow$ R sees one of the changes except with negl. prob.

Our Commitment Scheme, the implementation:

Pre-processing: independent of the input, public-key

- t-out-of-n OT on seeds $\{x_1, \ldots, x_n\}$ for a PRG [VZ12]
- Run the VSS with random strings $\{\mathbf{r}_1, \ldots, \mathbf{r}_\ell\}$ as input and send $row_i + PRG(x_i)$ for all i

On-line: field arithmetic, non-interactive

- Commit: Send $\mathbf{s} + \mathbf{r}_j$
- Reveal: Send all the shares for $\mathbf{r}_j$

Our Commitment Scheme: extensions

- additive homomorphism

  i.e. given $\mathbf{c}$ (commitment to $\mathbf{s}$), $\mathbf{c}'$ (commitment to $\mathbf{s}'$)

  $\mathbf{c} + \mathbf{c}'$ is a commitment to $\mathbf{s} + \mathbf{s}'$

Our Commitment Scheme: extensions

- additive homomorphism

  i.e. given $\mathbf{c}$ (commitment to $\mathbf{s}$), $\mathbf{c}'$ (commitment to $\mathbf{s}'$)

  $\mathbf{c} + \mathbf{c}'$ is a commitment to $\mathbf{s} + \mathbf{s}'$

- the receiver can check multiplicative relations

  i.e. given $\mathbf{c}$ (commitment to $\mathbf{s}$), $\mathbf{c}'$ (commitment to $\mathbf{s}'$) and $\mathbf{d}$

  check that $\mathbf{d}$ is a commitment to $\mathbf{s} \cdot \mathbf{s}'$

Our Commitment Scheme: extensions

- additive homomorphism

  i.e. given $\mathbf{c}$ (commitment to $\mathbf{s}$), $\mathbf{c}'$ (commitment to $\mathbf{s}'$)
  $\mathbf{c} + \mathbf{c}'$ is a commitment to $\mathbf{s} + \mathbf{s}'$

- the receiver can check multiplicative relations

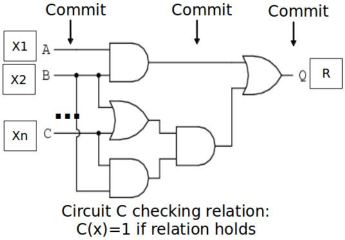
  i.e. given $\mathbf{c}$ (commitment to $\mathbf{s}$), $\mathbf{c}'$ (commitment to $\mathbf{s}'$) and $\mathbf{d}$
  check that $\mathbf{d}$ is a commitment to $\mathbf{s} \cdot \mathbf{s}'$

- given a commitment of $\mathbf{s} \longrightarrow$ compute a commitment for $\varphi(\mathbf{s})$
  (φ is a public linear map)

Our Commitment Scheme: applications

Efficient non-interactive UC **ZK proof of knowledge** for any NP relations [DIK10]

Prover



Verifier

- Verify relations between commitments;
- Check opening of commitment to output R;

if C is regular $\rightarrow O(|C|)$ complexity!

Recap:

We presented a **compact VSS** that:

- generalizes the construction of [CDM00] for packed LSSS;
- multiplication property and non-trivial extensions;
- constant communication rate;

The VSS scheme is used to design a **UC-commitment scheme** that:

- allows many commitments from a fixed number of seed OTs of fixed length and a PRG;
- non-interactive commit and open phases requiring only field arithmetic (linear complexity for the receiver!);
- additive and multiplicative homomorphism.

Thanks for your attention!