Zero Knowledge Proofs from Ring-LWE

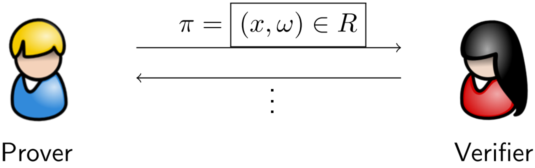
Xiang Xie, Rui Xue, Minqian Wang

Chinese Academy of Sciences

CANS 2013, Paraty

Outline

Zero-Knowledge Proofs [GoldwassorMicaliRackoff'85]



π reveals nothing except the statement itself.

Related Works of ZKPs

- Number Theoretical: [FeigeShamir'90], [CramerDamgård'98], [CramerDamgård'09], [GrothSahai'08] (paring), etc.
- General: [IshaiKushilevitzOstrovskySahai'07] (MPC).
- Lattice-Based: [MicciancioVadhan'03], [KawachiTanakaXagawa'08], [AsharovJainLópez-AltTromerVaikuntanathanWichs'12], [Lyubashevsky'08], [Lyubashevsky'12], [LingNguyenStehléWang'13].
- LPN-based: [JainKrennPietrzakTentes'12].

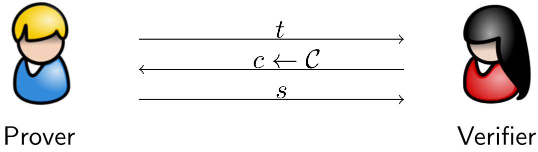
Our Results

- Commitment scheme from Ring Learning with Errors (RLWE).
- ZKP that proves the knowledge of the message hidden in our commitment scheme.
- Two ZKPs that prove component-wise relations of the messages in the commitment scheme.

Σ-Protocol

- Our ZKPs are essentially Σ-protocols (see [Damgård'04]).

Σ-protocol:



- **Completeness**: The verifier $\mathcal{V}$ accepts whenever $(x, \omega) \in \mathcal{R}$.
- **Special Soundness**: There exists a PPT algorithm Ext such that: $\omega' \leftarrow \mathsf{Ext}(\{(t, c, s_c) : c \in \mathcal{C}\})$, and $(x, \omega') \in \mathcal{R}$.
- **Special honest-verifier zero-knowledge**: There exists a PPT simulator S such that: $(t_x, c, s_x) \leftarrow \mathsf{S}(x, c) \approx (t, c, s)$.

- **Completeness**: The verifier $\mathcal{V}$ accepts whenever $(x, \omega) \in \mathcal{R}$.
- **Special Soundness**: There exists a PPT algorithm Ext such that: $\omega' \leftarrow \text{Ext}(\{(t, c, s_c) : c \in \mathcal{C}\})$, and $(x, \omega') \in \mathcal{R}$.
- **Special honest-verifier zero-knowledge**: There exists a PPT simulator S such that: $(t_x, c, s_x) \leftarrow \mathsf{S}(x, c) \approx (t, c, s)$.

Note:

- Σ-protocol can be extended to a ZKP for the same relation [Damgård'04], [DamgårdGoldreichOkamoto'95].
- Soundness is different from standard definition. We require Ext has input (t, c, s_c) for all $c \in \mathcal{C}$ with the same t. The knowledge error of the resulting ZKP scheme is $1 - 1/|\mathcal{C}|$ instead of $1/|\mathcal{C}|$.

Learning with Errors over Rings (RLWE)

- RLWE is introduced by Lyubashevsky, Peikert and Regev [LPR'10].

Let $R = \mathbb{Z}[X]/(X^d + 1)$, where $d = 2^k$ for some $k \geq 0$. For an integer q, let $R_q = R/qR$. The following two distributions are hard to distinguish:

$$\begin{array}{ll} a_1 \leftarrow R_q; & b_1 = a_1 \cdot s + e_1 \mod q \\ a_2 \leftarrow R_q; & b_2 = a_2 \cdot s + e_2 \mod q \\ \vdots & \\ a_m \leftarrow R_q; & b_m = a_m \cdot s + e_m \mod q \end{array}$$

$$\begin{array}{ll} a_1 \leftarrow R_q; & b_1 \leftarrow R_q \\ a_2 \leftarrow R_q; & b_2 \leftarrow R_q \\ \vdots & \\ a_m \leftarrow R_q; & b_m \leftarrow R_q \end{array}$$

Where $s \leftarrow R_q$, and $e_i \leftarrow \chi$ over R. $\|e_i\|_\infty \leq \beta \ll q$.

[LyubashevskyPeikertRegev'10]

If there exists a PPT algorithm solves RLWE problem, then there exists a PPT *quantum* algorithm solves some hard lattice problems for *all* d-dimensional *ideal lattices*.

Commitment from RLWE

The message space is R_q^ℓ. Let χ be a β-bounded distribution over R.

- KeyGen(1^λ) : Sample $\mathbf{a}_1 \leftarrow R_q^m$ and $\mathbf{A}_2 \leftarrow R_q^{m\times\ell}$, output $\mathbf{A} = [\mathbf{a}_1|\mathbf{A}_2] \in R_q^{m\times(\ell+1)}$.
- Com$(\mathbf{A}, \mathbf{m} \in R_q^\ell)$: Sample $s \leftarrow R_q$ and $\mathbf{e} \leftarrow \chi^m$, output $\mathbf{c} = \mathbf{A}[s|\mathbf{m}] + \mathbf{e} \in R_q^m$.
- Ver$(\mathbf{A}, \mathbf{c}, (s, \mathbf{m}))$: Accept iff $\|\mathbf{c} - \mathbf{A}[s|\mathbf{m}]\|_\infty \leq \beta$.

Commitment from RLWE

The message space is R_q^ℓ. Let χ be a β-bounded distribution over R.

- KeyGen(1^λ) : Sample $\mathbf{a}_1 \leftarrow R_q^m$ and $\mathbf{A}_2 \leftarrow R_q^{m\times\ell}$, output $\mathbf{A} = [\mathbf{a}_1|\mathbf{A}_2] \in R_q^{m\times(\ell+1)}$.
- Com$(\mathbf{A}, \mathbf{m} \in R_q^\ell)$: Sample $s \leftarrow R_q$ and $\mathbf{e} \leftarrow \chi^m$, output $\mathbf{c} = \mathbf{A}[s|\mathbf{m}] + \mathbf{e} \in R_q^m$.
- Ver$(\mathbf{A}, \mathbf{c}, (s, \mathbf{m}))$: Accept iff $\|\mathbf{c} - \mathbf{A}[s|\mathbf{m}]\|_\infty \leq \beta$.

Security:

- Computational hiding:

$$\mathbf{c} = \mathbf{A}[s|\mathbf{m}] + \mathbf{e} = \boxed{\mathbf{a}_1 \cdot s + \mathbf{e}} + \mathbf{A}_2\mathbf{m}$$

- Perfect binding: For uniformly random $\mathbf{A}$,

$$\Pr[\|\mathbf{y}\|_\infty \leq 2\beta : \mathbf{y} = \mathbf{A}\mathbf{x}, \mathbf{x} \neq \mathbf{0}] \leq \mathsf{negl}(\lambda).$$

Proving Knowledge of Valid Opending

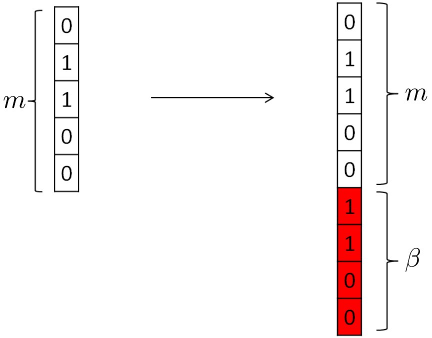
Relation:

$$\mathcal{R}_{\text{RLWE}} = \{((\mathbf{A}, \mathbf{c}), (s, \mathbf{m}, \mathbf{e})) \;:\; \mathbf{c} = \mathbf{A}(s\|\mathbf{m}) + \mathbf{e} \mod q \wedge \|\mathbf{e}\|_\infty \leq \beta\}.$$

- Extend Stern's ZKP for syndrome decoding problem. Similar to [JainKrennPietrzakTentes'12] and [LingNguyenStehléWang'13].
- The challenge set $\mathcal{C} = \{1, 2, 3\}$. The first two openings prove $\mathbf{A}, \mathbf{c}$ have the form $\mathbf{c} = \mathbf{A}[s|\mathbf{m}] + \mathbf{e}$.
- Obstacle: How to prove $\mathbf{e}$ is "short" without revealing anything else?

- If $\mathbf{e} \in \{0,1\}^m$ and $\|\mathbf{e}\|_1 = \beta$: Prover sends $\pi(\mathbf{e})$ for a uniformly random permutation π. $\pi(\mathbf{e})$ only reveals the Hamming weight of $\mathbf{e}$.

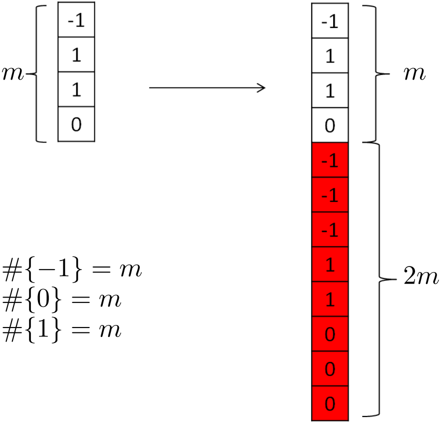
- If $\mathbf{e} \in \{0,1\}^m$ and $\|\mathbf{e}\|_1 = \beta$: Prover sends $\pi(\mathbf{e})$ for a uniformly random permutation π. $\pi(\mathbf{e})$ only reveals the Hamming weight of $\mathbf{e}$.
- If $\mathbf{e} \in \{0,1\}^m$ and $\|\mathbf{e}\|_1 \leq \beta$: Extend $\mathbf{e} \in \{0,1\}^m$ to $\mathbf{e}' \in \{0,1\}^{m+\beta}$ by padding, such that $\|\mathbf{e}'\|_1 = \beta$. Prover sends $\pi(\mathbf{e}')$.



- If $\mathbf{e} \in \mathbb{Z}^m$ and $\|\mathbf{e}\|_\infty \leq \beta$: Decompose $\mathbf{e}$:

$$\mathbf{e} = \sum_{i=0}^{k-1} 2^i \cdot \tilde{\mathbf{e}}_i, \; k = \lfloor \log \beta \rfloor + 1, \; \tilde{\mathbf{e}}_i \in \{-1, 0, 1\}^m$$

Extend $\tilde{\mathbf{e}}_i \in \{-1, 0, 1\}^m$ to $\mathbf{e}_i \in \{-1, 0, 1\}^{3m}$. Prover sends $\pi_i(\mathbf{e}_i)$.



$$\#\{-1\} = m$$
$$\#\{0\} = m$$
$$\#\{1\} = m$$

- If $\mathbf{e} \in R^m$ and $\|\mathbf{e}\|_\infty \leq \beta$. View $\mathbf{e} \in \mathbb{Z}^{dm}$ by the coefficient representation. The same as above.

Basic ZKP

Relation:

$$\mathcal{R}_{\text{RLWE}} = \{((\mathbf{A}, \mathbf{c}), (s, \mathbf{m}, \mathbf{e})) \ : \ \mathbf{c} = \mathbf{A}(s \| \mathbf{m}) + \mathbf{e} \mod q \wedge \|\mathbf{e}\|_\infty \leq \beta\}.$$

- Prover first decomposes $\mathbf{e} \in R^m$ to $\mathbf{e}_i \in R^{3m}$ according the method above.
- Define matrix $\hat{\mathbf{I}} = [\mathbf{I}_m | \mathbf{0}_m | \mathbf{0}_m] \in R^{m \times 3m}$.

Note that :

$$\mathbf{c} = \mathbf{A}(s|\mathbf{m}) + \mathbf{e} \Leftrightarrow \mathbf{c} = \mathbf{A}(s|\mathbf{m}) + \hat{\mathbf{I}}(\sum_{i=0}^{k-1} 2^i \cdot \mathbf{e}_i)$$

- Prover samples $(\mathbf{r}_0, ..., \mathbf{r}_{k-1}) \leftarrow (R_q^{3m})^k$, $\mathbf{v} \leftarrow R_q^{1+\ell}$, and k random permutations $(\pi_0, ..., \pi_{k-1})$. Sends:

$$\begin{cases} C_1 = & \mathsf{Com}\left(\{\pi_i\}_{i=0}^{k-1}, \mathbf{t}_1 = \mathbf{A}\mathbf{v} + \hat{\mathbf{I}}(\sum_{i=0}^{k-1} 2^i \cdot \mathbf{r}_i)\right) \\ C_2 = & \mathsf{Com}\left(\{\mathbf{t}_{2i} = \pi_i(\mathbf{r}_i)\}_{i=0}^{k-1}\right) \\ C_3 = & \mathsf{Com}\left(\{\mathbf{t}_{3i} = \pi_i(\mathbf{r}_i + \mathbf{e}_i)\}_{i=0}^{k-1}\right) \end{cases}$$

- Prover samples $(\mathbf{r}_0, ..., \mathbf{r}_{k-1}) \leftarrow (R_q^{3m})^k$, $\mathbf{v} \leftarrow R_q^{1+\ell}$, and k random permutations $(\pi_0, ..., \pi_{k-1})$. Sends:

$$\begin{cases} C_1 = & \mathsf{Com}\left(\{\pi_i\}_{i=0}^{k-1}, \mathbf{t}_1 = \mathbf{A}\mathbf{v} + \hat{\mathbf{I}}(\sum_{i=0}^{k-1} 2^i \cdot \mathbf{r}_i)\right) \\ C_2 = & \mathsf{Com}\left(\{\mathbf{t}_{2i} = \pi_i(\mathbf{r}_i)\}_{i=0}^{k-1}\right) \\ C_3 = & \mathsf{Com}\left(\{\mathbf{t}_{3i} = \pi_i(\mathbf{r}_i + \mathbf{e}_i)\}_{i=0}^{k-1}\right) \end{cases}$$

- Verifier chooses $Ch \leftarrow \{1, 2, 3\}$ and sends to Prover.

- Prover samples $(\mathbf{r}_0, ..., \mathbf{r}_{k-1}) \leftarrow (R_q^{3m})^k$, $\mathbf{v} \leftarrow R_q^{1+\ell}$, and k random permutations $(\pi_0, ..., \pi_{k-1})$. Sends:

$$\begin{cases} C_1 = & \text{Com}\left(\{\pi_i\}_{i=0}^{k-1}, \mathbf{t}_1 = \mathbf{A}\mathbf{v} + \hat{\mathbf{I}}(\sum_{i=0}^{k-1} 2^i \cdot \mathbf{r}_i)\right) \\ C_2 = & \text{Com}\left(\{\mathbf{t}_{2i} = \pi_i(\mathbf{r}_i)\}_{i=0}^{k-1}\right) \\ C_3 = & \text{Com}\left(\{\mathbf{t}_{3i} = \pi_i(\mathbf{r}_i + \mathbf{e}_i)\}_{i=0}^{k-1}\right) \end{cases}$$

- Verifier chooses $Ch \leftarrow \{1, 2, 3\}$ and sends to Prover.
- According to Ch, Prover does the following:

$$\begin{cases} Ch = 1, & \text{open } C_1, C_2; \\ Ch = 2, & \text{open } C_1, C_3; \\ Ch = 3, & \text{open } C_2, C_3. \end{cases}$$

- Prover samples $(\mathbf{r}_0, ..., \mathbf{r}_{k-1}) \leftarrow (R_q^{3m})^k$, $\mathbf{v} \leftarrow R_q^{1+\ell}$, and k random permutations $(\pi_0, ..., \pi_{k-1})$. Sends:

$$\begin{cases} C_1 = & \mathsf{Com}\Big(\{\pi_i\}_{i=0}^{k-1}, \mathbf{t}_1 = \mathbf{A}\mathbf{v} + \hat{\mathbf{I}}(\sum_{i=0}^{k-1} 2^i \cdot \mathbf{r}_i)\Big) \\ C_2 = & \mathsf{Com}\Big(\{\mathbf{t}_{2i} = \pi_i(\mathbf{r}_i)\}_{i=0}^{k-1}\Big) \\ C_3 = & \mathsf{Com}\Big(\{\mathbf{t}_{3i} = \pi_i(\mathbf{r}_i + \mathbf{e}_i)\}_{i=0}^{k-1}\Big) \end{cases}$$

- Verifier chooses $Ch \leftarrow \{1, 2, 3\}$ and sends to Prover.
- According to Ch, Prover does the following:

$$\begin{cases} Ch = 1, & \text{open } C_1, C_2; \\ Ch = 2, & \text{open } C_1, C_3; \\ Ch = 3, & \text{open } C_2, C_3. \end{cases}$$

- Verifier checks the following:

$$\begin{cases} Ch = 1, & \text{check } \mathbf{t}_1 - \hat{\mathbf{I}} \cdot \big(\sum_{i=0}^{k-1} 2^i \cdot \pi_i^{-1}(\mathbf{t}_{2i})\big) \in \mathsf{Img}(\mathbf{A}); \\ Ch = 2, & \text{check } \mathbf{t}_1 + \mathbf{c} - \hat{\mathbf{I}} \cdot \big(\sum_{i=0}^{k-1} 2^i \cdot \pi_i^{-1}(\mathbf{t}_{3i})\big) \in \mathsf{Img}(\mathbf{A}); \\ Ch = 3, & \text{check } \mathbf{t}_{3i} - \mathbf{t}_{2i} \in \{-1, 0, 1\}^{3md}. \end{cases}$$

- Correctness: obvious.
- Special Soundness: $Ch = 1$ and $Ch = 2$ guarantee that $\mathbf{A}, \mathbf{c}$ have the proper form. $Ch = 3$ guarantees $\mathbf{e}$ is small.
- Special Honest-Verifier Zero-Knowledge: By the decomposition and extension technique. Similar to [LingNguyenStehléWang'13].

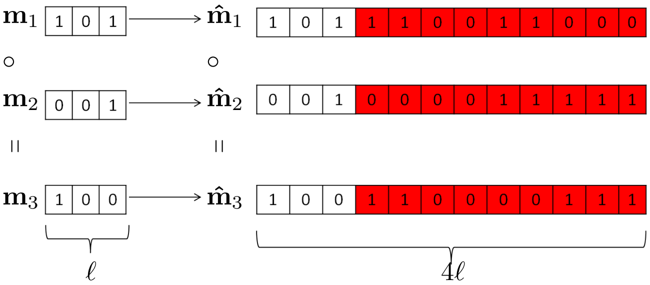
Component-Wise Relations

Relation:

$$\mathcal{R}_{\mathsf{CWRLWE}} = \Big\{ \Big((\mathbf{A}, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3), (s_1, s_2, s_3, \mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) \Big) :$$
$$\bigwedge_{i=1}^{3} \Big(\mathbf{c}_i = \mathbf{A}(s_i|\mathbf{m}_i) + \mathbf{e}_i \mod q \wedge \|\mathbf{e}_i\|_\infty \leq \beta \Big) \wedge \mathbf{m}_3 = \mathbf{m}_1 \circ \mathbf{m}_2 \Big\}.$$

Where $\circ$ denotes the component-wise addition or multiplication in R_q.

- If $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3 \in \{0,1\}^\ell$, extend them to $\hat{\mathbf{m}}_1, \hat{\mathbf{m}}_2, \hat{\mathbf{m}}_3$



$$\#\{(1,1)\} = \#\{(1,0)\} = \#\{(0,1)\} = \#\{(0,0)\} = \ell$$

- Prover sends $\pi(\hat{\mathbf{m}}_1), \pi(\hat{\mathbf{m}}_2), \pi(\hat{\mathbf{m}}_3)$, note that

$$\pi(\hat{\mathbf{m}}_1) \circ \pi(\hat{\mathbf{m}}_2) = \pi(\hat{\mathbf{m}}_3)$$

- If $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3 \in \mathbb{Z}_q^\ell$. Extend to $\hat{\mathbf{m}}_1, \hat{\mathbf{m}}_2, \hat{\mathbf{m}}_3 \in \mathbb{Z}_q^{q^2\ell}$ as before. Note: This method only works for $q = \mathsf{poly}(\lambda)$.

- If $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3 \in \mathbb{Z}_q^{\ell}$. Extend to $\hat{\mathbf{m}}_1, \hat{\mathbf{m}}_2, \hat{\mathbf{m}}_3 \in \mathbb{Z}_q^{q^2 \ell}$ as before. Note: This method only works for $q = \text{poly}(\lambda)$.
- If $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3 \in R_q^{\ell}$. The dimension of $\hat{\mathbf{m}}_1, \hat{\mathbf{m}}_2, \hat{\mathbf{m}}_3$ is exponential!!!

How to overcome this problem ?

Chinese Remainder Theorem (CRT)

Let $R = \mathbb{Z}[X]/(X^d + 1)$ with $d = 2^k$ for $k \in \mathbb{N}^+$. Let $R_q = R/qR$.

- Coefficient Representation: For $a \in R_q$, i.e $a(X) = \sum_{i=0}^{d-1} a_i X^i$. Represent a by

$$(a_0, ...a_d) \in \mathbb{Z}_q^d.$$

- CRT Representation: If $q = 1 \mod 2d$ and q is prime, then

$$(X^d + 1) = \prod_{i=1}^{d} (X - \zeta_i) \mod q.$$

Represent a by

$$\left(a(\zeta_1), ..., a(\zeta_d)\right) \in \mathbb{Z}_q^d.$$

- Add: for $a, b \in R_q$, then $a + b$ is

$$\Big(a(\zeta_1) + b(\zeta_1), ..., a(\zeta_d) + b(\zeta_d)\Big) \in \mathbb{Z}_q^d.$$

- Multiplication: for $a, b \in R_q$, then $a \cdot b \in R_q$ is

$$\Big(a(\zeta_1)b(\zeta_1), ..., a(\zeta_d)b(\zeta_d)\Big) \in \mathbb{Z}_q^d.$$

We now can adapt the extension technique in the CRT representation.

Reduce Communication Complexity

- The method extends the dimension from ℓ to $q^2\ell$. Large Communication Complexity !
- The reason is that we consider multiplication in $\mathbb{Z}_q$. We note that for addition, there is much simpler methods (without extension) due to the linearity.

Reduce Communication Complexity

- The method extends the dimension from ℓ to $q^2\ell$. Large Communication Complexity !
- The reason is that we consider multiplication in $\mathbb{Z}_q$. We note that for addition, there is much simpler methods (without extension) due to the linearity.
- When proving multiplication, instead of directly extend the vector, we consider the following relations:

$$\mathbf{m}_1 = \sum_{j=0}^{\lfloor \log q \rfloor} 2^j \cdot \mathbf{m}_{1j}; \quad \mathbf{m}_2 = \sum_{k=0}^{\lfloor \log q \rfloor} 2^k \cdot \mathbf{m}_{2k};$$

$$\mathbf{m}_{jk} = \mathbf{m}_{1j} \diamond \mathbf{m}_{2k}; \qquad \mathbf{m}_3 = \sum_{j,k} 2^{j+k} \cdot \mathbf{m}_{jk}.$$

$\diamond$ means component-wise bit multiplication.

Reduce Communication Complexity

- The method extends the dimension from ℓ to $q^2\ell$. Large Communication Complexity !
- The reason is that we consider multiplication in $\mathbb{Z}_q$. We note that for addition, there is much simpler methods (without extension) due to the linearity.
- When proving multiplication, instead of directly extend the vector, we consider the following relations:

$$\mathbf{m}_1 = \sum_{j=0}^{\lfloor \log q \rfloor} 2^j \cdot \mathbf{m}_{1j}; \quad \mathbf{m}_2 = \sum_{k=0}^{\lfloor \log q \rfloor} 2^k \cdot \mathbf{m}_{2k};$$

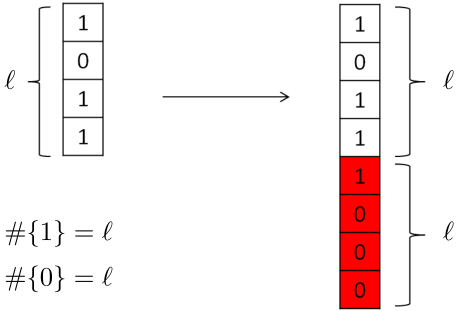
$$\mathbf{m}_{jk} = \mathbf{m}_{1j} \diamond \mathbf{m}_{2k}; \qquad \mathbf{m}_3 = \sum_{j,k} 2^{j+k} \cdot \mathbf{m}_{jk}.$$

$\diamond$ means component-wise bit multiplication.

Note: we only need to extend the dimension to prove $\mathbf{m}_{jk} = \mathbf{m}_{1j} \diamond \mathbf{m}_{2k}$. Since they are all binary vectors, the dimension is extended from ℓ to 4ℓ. But the prover needs extra $\log^2 q + \log q$ commitments.

- Be Careful: Prover must convince Verifier that $\mathbf{m}_{1j}$ and $\mathbf{m}_{2k}$ are bit vectors.

- Be Careful: Prover must convince Verifier that $\mathbf{m}_{1j}$ and $\mathbf{m}_{2k}$ are bit vectors.
- To prove $\mathbf{m} \in \{0,1\}^{\ell}$. Prover extends $\mathbf{m}$ to $\bar{\mathbf{m}} \in \{0,1\}^{2\ell}$ and sends $\pi(\bar{\mathbf{m}})$,



- We now can prove any polynomial relations of the messages under the commitment.
- The amortized complexity is $\tilde{O}(\lambda|f|)$, where f is the polynomial relation.

Questions ?