Efficient Conversion of Secret-shared Values Between Different Fields

Ivan Damgård and Rune Thorbek

BRICS, Dept. of Computer Science, University of Aarhus

Abstract. We show how to effectively convert a secret-shared bit b over a prime field to another field. If initially given a random replicated secret share this conversion can be done by the cost of revealing one secret shared value. By using a pseudo-random function it is possible to convert arbitrary many bit values from one initial random replicated share. Furthermore, we generalize the conversion to handle values.

1 Introduction

In secure multi-party computation where n players what to distributed compute some pre-determined function $f(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$ privately it is often beneficial to do some computations on bits. For instance, Damgård et al. showed in [2] how some operations can be efficiently done over bit-sharings.

When efficiency is considered in bit-arithmetic the operation *xor* is crucial. If this operation is done on secret shared bits over a field of characteristic $p \neq 2$ it costs one multiplication. While on the other hand, if it is done in a field of characteristic 2 it can be done without interaction. Most often, the obtained bits are in a prime field $\mathbb{Z}_p$. Converting a shared bit b over $\mathbb{Z}_p$ to a field of characteristic 2 is therefore very useful.

To our knowledge, the previous best solution to solve the conversion problem, is to let each player secret share a random number r_i over $\mathbb{Z}_p$ and share the least significant bit of r_i over a field of characteristic 2, called the *target field*. Furthermore, the random number r_i needs to be bounded by $\lfloor \log(p) \rfloor - \lfloor \log(n) \rfloor$ bits. This ensures, that the sum of all the players random shared values $r = \sum_i r_i$ will not lead to a modular reduction. Hence, the sum of all the least significant bits in the target field will be the least significant bit of r. When this is done, the actual conversion of the secret shared bit b over the prime field, is done by opening $r+b$ and use the opened value to adjust the shared bit in target field. Observe that this protocol only handles the case of a passive adversary which follows the protocol. For an active adversary, which may deviate from the protocol, each player needs to prove that the least significant bit is correct and that the random value is not too big.

In this paper we present a novel technique to solve this problem. Under the assumption that the players have secret shared a random replicated value, this can be done only by opening one value. In the threshold case with no more than t corrupted players the predistributed replicated share consists of $\binom{n}{t}$ elements, where n is the number of players. This might seem as a big value, but for small n this is still acceptable. Furthermore, in the case of a $t < n/3$ threshold, the protocol is secure against an active adversary.

Furthermore, the costs of the initial replicated secret share can be further reduced if assuming that pseudo-random functions (PRF) are secure. If the players jointly generate one public random value, used as a seed for PRF, then the same random replicated secret share can be used.

2 Preliminaries

Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ denote n players and let Γ be a set of subsets of $\mathcal{P}$.

Definition 1. *Γ is called an* monotone access structure *over $\mathcal{P}$, if it is closed under supersets, i.e., if $A \in \Gamma, B \supset A \Rightarrow B \in \Gamma$.*

Note 1. The threshold access structure $T_{t,n} = \{A \subseteq \mathcal{P} \mid |A| > t\}$ is a special case of a monotone access structure.

The sets in a monotone access structure Γ over $\mathcal{P}$ are called *qualified.*

Given a monotone access structure Γ, let $\mathcal{T}$ denote all the *maximal* unqualified sets of Γ. In a replicated secret-sharing scheme [4] a random number r_T for each set $T \in \mathcal{T}$ is chosen, with the restriction that $s = \sum_{T \in \mathcal{T}} r_T$ equals the secret-shared value. Then each share r_T is given to all players $P_j \notin T$. That is, the share of player P_j is $\{r_T\}_{P_j \notin T}$.

Privacy follows from the fact that members of every maximal unqualified set $T \in \mathcal{T}$ jointly miss exactly one additive share, namely the share r_T. Since Γ is monotone, a qualified set $Q \in \Gamma$ will jointly view all shares r_T and can thus reconstruct s.

Definition 2 (Share Conversion [1]). *Let S, S' be two secret-sharing schemes over the same secret-domain K. We say that S is* locally convertible *to S' if there exist local conversion functions $g_1, \ldots, g_n$ such that the following holds. If $(s_1, \ldots, s_n)$ are valid shares of a secret s in S, then $(g_1(s_1), \ldots, g_n(s_n))$ are valid shares of the same secret s in S'.*

In [1] Cramer et al. show how to locally convert a replicated secret-share s to a Shamir shares [5]. First note that the maximal unqualified sets of the $T_{t,n}$ threshold structure are given by $\mathcal{T} = \{A \subseteq \mathcal{P} \mid |A| = t\}$. Thus, we have

$$s = \sum_{A \in \mathcal{T}} r_A,$$

where r_A has been given to all players not in A.

Furthermore, let x_i be a unique point assigned to P_i. For each $A \in \mathcal{T}$, let f_A be the degree-t polynomial such that:

$$f_A(0) = 1,$$

and

$$f_A(x_i) = 0 \quad \forall P_i \in A.$$

Every player P_j can compute a share s_j as follows:

$$s_j = \sum_{A \in \mathcal{T}} r_A \cdot f_A(x_j),$$

which defines consistent shares of the polynomial

$$f = \sum_{A \in \mathcal{T}} r_A \cdot f_A.$$

First observe that by the locality of the above conversion, converted shares cannot reveal more information about the secret then the original shares.

Furthermore observe that the functions $f_A \in \mathbb{F}[X]$ and the values r_A might not be from the same domain. Say, if $r_A \in \mathbb{Z}$ and $f_A \in \mathbb{Z}_p[X]$, then the above conversions still is meaningful, since the map $\cdot : \mathbb{Z} \times \mathbb{Z}_p[X] \rightarrow \mathbb{Z}_p[X]$ can be seen as a $\mathbb{Z}$-module. In this case, the replicated share $s = \sum r_A$ will be locally converted to a Shamir sharing of s mod p. Furthermore, note if the functions were chosen from $GF(2^l)[X]$, i.e., polynomials over a field of characteristic 2, the locally conversion will result in a Shamir sharing of s mod 2.

The replicated secret sharing scheme from [4] is done over some field, but Damgård and Thorbek showed in [3] how the replicated secret-sharing scheme from [4] can be realized over the integers and proved secure.

First we formally define the scheme over the integers.

Scheme Replicated Integer Secret-Sharing (RISS) R_{Γ}

Let $\mathcal{T}$ be the maximal unqualified sets of an access structure Γ. For each set $T \in \mathcal{T}$ choose an uniformly random r_T integer from the interval $[-2^{l+k}..2^{l+k}]$ and send privately r_T to each player $P_i \notin T$. Furthermore, publish $r = s + \sum_{T \in \mathcal{T}} r_T$, where s is the secret from the interval $[-2^l..2^l]$.

Note that the public value $r = s + \sum_{T \in \mathcal{T}} r_T$ does not further complicate the conversion. Simply follow the protocol above, resulting in a polynomial $f \in \mathbb{F}[X]$ such that $f(0) = \sum r_T \in \mathbb{F}$. Then using the constant polynomial $g(x) = s + \sum r_T$ to locally obtain sharings of $F = g - f$, i.e., $F(0) = s$.

Note that even though the obtained Shamir sharing is well known to be secure, it does not necessarily mean that the RISS sharing is secure, or more precisely, correct and private (defined below).

Definition 3. *A RISS scheme is* correct*, if the secret can be reconstruct from shares of any qualified set in $A \in \Gamma$, by taking an integer linear combination of the shares with coefficient that depends only on the index set A.*

Definition 4. *A RISS scheme is* private*, if for any forbidden set $B \notin \Gamma$, any two secret $s, s' \in [-2^l..2^l]$, and independent random coins r and r', the statistical distance between the distributions of the shares $\{s_i(s, r, k) \mid i \in B\}$ and $\{s_i(s', r', k) \mid i \in B\}$ is negligible in the security parameter k.*

Lemma 1 ([3]). *The RISS scheme is correct and private.*

As stated above, given a replicated share for a threshold structure it can be locally converted to a Shamir share, where the conversion is done as a $\mathbb{Z}$-module. Actually, a more general result was proved in [3].

Proposition 1 ([3]). *The RISS scheme R_Γ, realizing Γ, is locally convertible to any LSS over $\mathbb{Z}_p$ realizing an access structure $\Gamma' \subseteq \Gamma$, where the original secret s after conversion will be $s \bmod p$.*

Furthermore, as noted above, if the target field of the share conversion has characteristic 2, then the resulting share will be the least significant bit of the RISS share.

3 Bit Conversion

In this section we will show how the replicated integer secret-sharing (RISS) can be used to convert a secret shared bit over a prime-field to a field of characteristic 2 as well as to an arbitrary field.

First we introduce some notation. Let $[a]$ denote a RISS sharing of a, that is, it represents $\{r_A\}$ such that $a = \sum r_A$. Let $[a]_p$ denote a Shamir share over $\mathbb{Z}_p$, and finally we denote $[a]_2$ as a Shamir share over a field of characteristic 2.

3.1 Conversion to a Field of Characteristic 2

Assume that the players have shared a RISS share $\{r_A\}$ for a threshold structure among them, with the restriction that all $0 \leq r_A < 2^l$ for all $A \in \mathcal{T}$. Choose p such that $p > r = \sum r_A$.

Then the protocol is as follows. Given $[b]_p$ for some bit b the goal is to convert it to $[b]_2$. By assumption we are given $[r]$ which can be locally converted to $[r]_p$ and $[r_0]_2$, where r_0 denotes the least significant bit of r. This is possible since $0 \leq r = \sum r_A < p$. First note, that the polynomials in the conversion to $[r_0]_2$ are taken from a field of characteristic 2, hence the share will be either 0 or 1, as earlier noted. Furthermore, since the least significant bit of $r \bmod p$ equals the least significant bit of $r \in \mathbb{Z}$ it follows.

Then open the value $[r+b]_p = [r]_p + [b]_p$ to reveal $r+b$. Note that the least significant bit of $(r+b)$ is $r_0 \oplus b$. Use this value to modify $[r_0]_2$ to obtain $[b]_2$.

Note that the only information leaked in the protocol is $(r+b)$ which statistically hides the bit b in the parameter l.

3.2 Conversion Between Two Prime Fields

Note, that there is nothing in the protocol from the above section that makes it necessary to restrict the target field to have characteristic 2.

Starting from $[r]$ obtain $[r]_p$ and $[r]_q$, where we still have that $p > r$ and the replicated shares are positive and bounded by 2^l. Then open the value $[r+b]_p$ and use $r + b \bmod q$ to modify $[r]_q$ to $[b]_q$.

Finally note, that we do not need to restrict the conversion to a bit b, we only need to require that the converted value is statistically hidden by the random value r. That is, if we want to convert a value $a \in [0..2^l]$ then we need r to be random in $[0..2^{l+k}]$ for some security parameter k that determines the statistical leakage.

3.3 Multiple Conversions

Given random keys $\{r_A\}$ for a RISS share $[r]$ and a pseudo-random function (PRF) $\psi : \mathbb{Z} \times \mathbb{Z} \rightarrow [0..2^l - 1]$ it is possible to make arbitrary many conversions as follows. The players distributed agree upon a value a and use the keys $\{\psi_{r_A}(a)\}$ as the initial RISS share. Otherwise proceed as the protocol above.

3.4 The Initial RISS Share

In the previous sections we have assumed that an initial random RISS share has been dealt among the players. This can be solved by letting each player share a random value in RISS and using the sum of those. This ensures that an adversary that corrupts all but one player, will still not be able to determine the secret, since at least one l-bit value remains unknown to him. Furthermore, the l-bit value is enough to statistically hide the converted bit or value.

If all player share an RISS value with l-bit share elements, the resulting share elements will be $l + \log(n)$-bits, where n is the number of players. This is not a problem, but should just be noted when choosing the size of the field to convert from.

4 Acknowledgments

We thank Tomas Toft and Martin Geisler for helpful discussions.

References

1. Ronald Cramer, Ivan Damgård, Yuval Ishai: *Share Conversion, Pseudorandom Secret-Sharing and Applications to Secure Computation.* Proc. of TCC 2005, pp. 342-362, Springer Verlag LNCS
2. Ivan Damgrd, Matthias Fitzi, Eike Kiltz, Jesper Buus Nielsen, Tomas Toft: *Unconditionally Secure Constant-Rounds Multi-party Computation for Equality, Comparison, Bits and Exponentiation.* Proc. of TCC 2006, pp. 285-304, Springer Verlag LNCS.
3. Ivan Damgård and Rune Thorbek *Non-Interactive Proofs for Integer Multiplication*, EUROCRYPT'07, LNCS 4514, pp 412-429, 2007.
4. M. Ito, A. Saito, and T. Nishizeki. *Secret sharing schemes realizing general access structures.* Proc. IEEE Global Telecommunication Conf., Globecom 87: 99-102 (1987).
5. Adi Shamir: *How to Share a Secret.* Commun. ACM 22(11), pp 612-613, 1979.