Braid Based Cryptosystems

Kate Berry

10 December 2009

Background on Braids

Definition: For $n \geq 2$, the braid group B_n is defined by:

$$\langle \sigma_1, \ldots, \sigma_{n-1} \, ; \, \sigma_i \sigma_j = \sigma_j \sigma_i \text{ for } |i-j| \geqslant 2, \, \sigma_i \sigma_j \sigma_i = \sigma_j \sigma_i \sigma_j \text{ for } |i-j| = 1 \rangle.$$

For each n, the identity mapping embeds B_n into B_{n+1} so that the groups B_n arrange into a more complex grouping

Each σ_i can be seen as a projection of a three dimensional figure

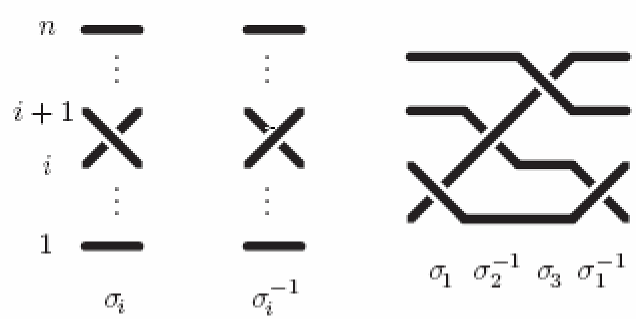

FIGURE 1. Braid diagrams associated with σ_i, σ_i^{-1}, and with $\sigma_1 \sigma_2^{-1} \sigma_3 \sigma_1^{-1}$

Background on Braids

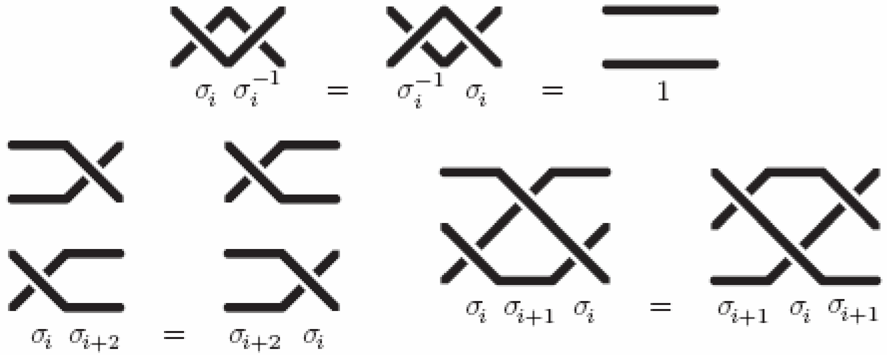

FIGURE 2. Geometric interpretation of the braid relations

Two braids *p, p'* are *conjugate* if *p'* = sps^{-1} for some braid *s*.

The *Conjugacy Problem* is the question of algorithmically recognizing whether two braids *p, p'* are conjugate

The *Conjugator Search Problem* is the related question of finding a conjugating braid for a pair (*p, p'*) of conjugate braids, *i.e.*, finding *s* satisfying *p'* = sps^{-1}.

Braid Based Key Exchange
The Anshel-Anshel-Goldfield Scheme

The public key consists of two sets of braids, $p_1,\ldots,p_l$, $q_1,\ldots,q_m$, in B_n.

Alice's secret key is a word u on l letters and their inverses

Bob's secret key is a word v on m letters and their inverses

- A computes the braid $s = u(p_1, \ldots, p_\ell)$, and uses it to compute the conjugates $q'_1 = sq_1s^{-1}, \ldots, q'_m = sq_ms^{-1}$; she sends $q'_1, \ldots, q'_m$;
- B computes the braid $r = v(q_1, \ldots, q_m)$, and uses it to compute the conjugates $p'_1 = rp_1r^{-1}, \ldots, p'_\ell = rp_\ell r^{-1}$; he sends $p'_1, \ldots, p'_\ell$;
- A computes $t_A = s\, u(p'_1, \ldots, p'_\ell)^{-1}$;
- B computes $t_B = v(q'_1, \ldots, q'_m)\, r^{-1}$.

The common key is $t_A = t_B$.

To check this, we can see that

$$t_A = s\, u(p'_1, \ldots, p'_\ell)^{-1} = s\, r\, u(p_1, \ldots, p_\ell)^{-1}\, r^{-1}$$
$$= s\, r\, s^{-1}\, r^{-1} = s\, v(q_1, \ldots, q_m)\, s^{-1}\, r^{-1} = v(q'_1, \ldots, q'_m) r^{-1} = t_B$$

Braid Based Key Exchange:
A Diffie-Hellman-like Scheme

Braids involving disjoint sets of strands commute.
Let LB_n the subgroup of Bn generated by $\sigma_1, \ldots, \sigma_{m-1}$ and UB_n generated by $\sigma_{m+1}, \ldots, \sigma_{n-1}$ with $m = n/2$,
Note that every braid in LB_n commutes with every braid in UB_n.

The public key consists of one braid p in B_n
Alice's secret key s is in LB_n and Bob's secret key r is in UB_n

- A computes the conjugate $p' = sps^{-1}$, and sends it to B;
- B computes the conjugate $p'' = rpr^{-1}$, and sends it to A;
- A computes $t_A = s\,p''\,s^{-1}$;
- B computes $t_B = r\,p'\,r^{-1}$.

The common key is $t_A = t_B$.

Thus because s and r commute, we have

$$t_A = s\,p''\,s^{-1} = s\,r\,p\,r^{-1}\,s^{-1} = r\,s\,p\,s^{-1}\,r^{-1} = r\,p'\,r^{-1} = t_B.$$

Authentication:
A Diffie-Hellman-like Scheme

The public key is a pair of conjugate braids (p, p') in B_n with $p'=sps^{-1}$,

Alice's private key is the braid s used to conjugate p into p'

s belongs in LB_n and h is a collision free, one way hash function on B_n

- B chooses a random braid r in UB_n, and he sends the challenge $p'' = rpr^{-1}$ to A;
- A sends the response $y = h(sp''s^{-1})$;
- B checks $y = h(rp'r^{-1})$.

the braids *r* and *s* commute so $rp'r^{-1} = sp''s^{-1}$.

Authentication:
A Fiat-Shamir-like Scheme

As before, the public keys are a pair of conjugate braids (*p, p'*) with *p'* = *sps*$^{-1}$, while *s*, the conjugating braid, is Alice's private key.

In contrast to the previous schemes, both *p* and *s* lie in B_n. We still assume that *h* is a collision-free one-way hash function on B_n. The authentication procedure consists in repeating *k* times the following three exchanges:

- A chooses a random braid r in B_n, and she sends the *commitment* $x = h(rp'r^{-1})$;
- B chooses a random bit c and sends it to A;
- For $c = 0$, A sends $y = r$, and B checks $x = h(yp'y^{-1})$;
- For $c = 1$, A sends $y = rs$, and B checks $x = h(ypy^{-1})$.

Braid Based Signature

The public keys are a pair of conjugate braids (*p, p'*) with $p' = sps^{-1}$, s is Alice's private key; the braids *p* and *s* belong to B_n.
We use *H* for a one-way collision-free hash function from $\{0, 1\}^*$ to B_n
we use ~ for conjugacy in B_n.
The first scheme is as follows:

- A signs the message m with $q' = sqs^{-1}$, where $q = H(m)$;
- B checks $q' \sim q$ and $p'q' \sim pq$.

A possible weakness of the previous scheme lies in that repeated uses disclose many conjugate pairs (q_i, q_i) associated with the common conjugator *s*. To avoid this, the scheme can be modified by incorporating an additional random braid.

- A chooses a random braid r in B_n;
- A signs the message m with the triple (p'', q'', q'), where $p'' = rpr^{-1}$, $q = H(mh(p''))$, $q'' = rqr^{-1}$, and $q' = rs^{-1}qsr^{-1}$;
- B checks $p'' \sim p$, $q'' \sim q' \sim q$, $p''q'' \sim pq$, and $p'q' \sim p'q$.

References

Dehornoy, Patrick. *Braid-based cryptography.* Contemporary Mathematics. *http://www.math.unicaen.fr/~dehornoy/Surveys/Dgw.pdf, 2004.*

Weisstein, Eric W. "Braid Group." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/BraidGroup.html