The Algebraic Eraser: a linear asymmetric protocol for low-resource environments

Derek Atkins, Paul E. Gunnells

SecureRF Corporation

IETF92 (3/25/15)

Algebraic Eraser

- I. Anshel, M. Anshel, D. Goldfeld, and S. Lemieux, *Key agreement, the Algebraic Eraser*$^{\mathrm{TM}}$*, and lightweight cryptography*, Algebraic methods in cryptography, Contemp. Math., vol. 418, Amer. Math. Soc., Providence, RI, 2006, pp. 1–34.
- Asymmetric key agreement protocol
- Designed for low-cost platforms with constrained computational resources
  - RFID
  - Bluetooth
  - NFC
  - "Internet of Things"
- Complexity scales *linearly* with desired security level, unlike RSA, ECC.

SECURE RF
Securing the Internet of Things

VERIDIFY

AE Performance vs ECC

2^{128} Security level (AES–128)

ECC 283			AE B_{16}, $\mathbb{F}_{256}$			Gain
Cycles	Gates	Wtd. Perf.	Cycles	Gates	Wtd. Perf.	
164,823	29,458	4,855,355,934				71.7x
85,367	77,858	6,646,503,866	3,352	20,206	67,730,512	98.1x
70,469	195,382	13,768,374,158				203.3x

Wtd. Perf. is Weighted Performance (clock cycles $\times$ gate count) and represents time and power usage. Gate counts are for 65nm CMOS. ECC data taken from *A Flexible Soft IP Core for Standard Implementations of Elliptic Curve Cryptography in Hardware*, B. Ferreira and N. Calazans, 2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS), 12/2013.

SECURE RF
Securing the Internet of Things

VERIDIFY

Overview of AE

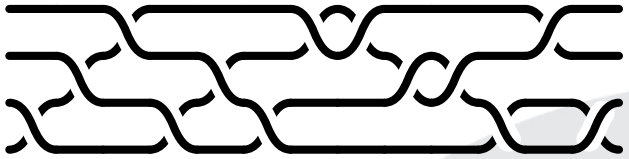
- The AE key exchange is a nonabelian Diffie–Hellman exchange.
- The underlying algebraic structure is not $(\mathbb{Z}/N\mathbb{Z})^\times$ or $E(\mathbb{F}_q)$, but rather
  - $M_n(\mathbb{F}_q)$ ($n \times n$ matrices over $\mathbb{F}_q$),
  - B_n (the braid group on n strands).
- Private keys: a pair $R = (m, \mu)$ of a matrix and braid.
- Public keys: a pair $P = (M, \sigma)$ of a matrix and a permutation in S_n.
- Each user also knows a fixed ordered list of elements of $\mathbb{F}_q$ (*T-values*).
- The shared secret: same kind of pair as the public key.

Overview of AE

- The security level depends on n, q and the lengths of the private braids (and scales linearly with the lengths of the braids).
- The (maximum) security level for AE is $n \cdot \lg q$, not $(\lg q)/2$ as in ECC. In particular one can use moderately sized finite fields, not multiprecision finite fields.
- The hard computational problem underlying AE takes place in the braid group B_n, and is known as the *Simultaneous conjugacy separation search problem*. This is not the same computational problem underlying earlier braid group schemes, and AE is not "Braid Group Cryptography."

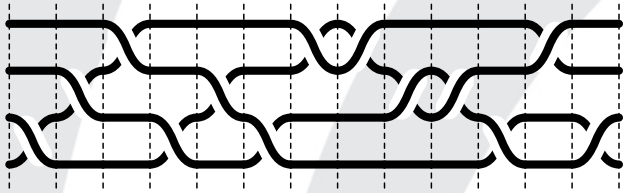
Braids

A braid on n strands is a collection of n entangled strings.



We can represent a braid by a *left-right crossing sequence* of signed nonzero integers $i_1 i_2 \cdots i_k$, ("Artin generators") each of which lies between $-n$ and n.

- A positive integer i means "cross the ith strand *under* the $(i+1)$st strand."
- A negative integer $-i$ means "cross the ith strand *over* the $(i+1)$st strand."



$$1\ 2\ 3\ 1\ 2\ 1\ 3\ -3\ -2\ -2\ 1\ -3\ -1$$

E-multiplication

E-multiplication is an action of B_n on $M_n(\mathbb{F}_q)$

- Each Artin generator determines an $n \times n$ sparse matrix, a *colored Burau matrix*.
- This matrix depends on the T-values (the fixed set of elements in $\mathbb{F}_q$), but the correspondence between generators and matrices changes as one moves down the braid in the private key.
- This nontrivial permuting of the T-values is the "eraser" part of the construction. Effectively it masks the map between braids and matrices.
- E-multiplication is how the public keys are produced from the private data: $P_A = m_A \star \mu_A$, $P_B = m_B \star \mu_B$ ($A =$ Alice, $B =$ Bob).

Shared secret computation

- Bob and Alice take each others public keys $P_A = (M_A, \sigma_A), P_B = (M_B, \sigma_B)$, and multiply their private matrices m_A, m_B against them.
- Then they E-multiply the result by their braids μ_A, μ_B:

$$S_A = P_B m_A \star \mu_A, \quad S_B = P_A m_B \star \mu_B.$$

- We have $S_A = S_B$.

Many details have of course been elided, for example how one chooses the matrices and braids.

Thank You!

SecureRF Corporation
100 Beard Sawmill Rd, Suite 350
Shelton, CT 06484

Derek Atkins (datkins@securerf.com)
Paul Gunnells (pgunnells@securerf.com)