CS 259C/Math 250: Elliptic Curves in Cryptography
Final Project

David Mandell Freeman

November 21, 2011

1 The Assignment

The final project is an expository paper that surveys some research issue relating to elliptic curves in cryptography. Specifically, you will read 2–3 papers on a subject and write a report that describes the general problem and some interesting aspect of the conceptual and technical contributions of the papers.

Your report definitely does not need to (and probably should not) cover all of the technical content of those papers; instead, just pick the most enlightening parts. Put some thought into how to strip the subject down into the most essential ideas to take away (as opposed to the boring technical calculations), and how to present them in a simplified form that will be easy to understand. (If you had to sit in on a 90-minute lecture, what would you most want to hear about?) Your report should make clear how the presented content fits into the broader field of elliptic curve cryptography. You may treat all of your papers equally, or you may choose to focus on results of one paper and use the others for background or comparison.

Your paper should contain the statement and proof of at least one key result in the papers you have read, possibly filling any gap in the original presentation. You should also describe clearly of how the presented result fits into the general picture. Your result could be:

- A purely mathematical statement: "If a widget is normal and symmetric, then it is also a gizmo."
- A statement about a cryptographic construction: "If the ABC assumption holds, then the PQR cryptosystem is secure."
- The correctness of an algorithm: "If algorithm A is given input x, y, it outputs either $\perp$ or $f(x, y)$. The algorithm runs in time $O(\log^3 x + y)$ and outputs $\perp$ with probability at most $1/8$."

Make sure that all of your terms are defined! If you were to include all of the above, you would have to define widget, normal, symmetric, and gizmo; say what the ABC assumption is, say what "secure" means for the PQR cryptosystem, and define the function f.

The expectation is that you should take effort to make your report "beautiful." Your report does not have to provide a comprehensive survey of the entire topic area you chose, but the parts it does present should be well thought-out, clearly presented and organized, conceptually cohesive, and easy to read.

If you're at a loss for a project topic, I have prepared a list of possible topics that you can peruse as examples of how to a pick a suitable project. (See below.) Don't feel limited to these suggestions — they are intended only as examples.

2 Technical Details

Page limit

There is no page limit (either minimum or maximum), and reports will be evaluated on technical content (not on length), but I expect that a typical report will be about 5 to 8 pages long.

Collaboration

Projects will be done individually.

Proposals

When you have chosen a project topic, please send email to dfreeman at cs.stanford.edu describing your project. The email should contain (either in the body or in an attached text or pdf file):

- The title of the project.
- A short description of the topic (2–3 paragraphs).
- A list of the papers you are planning to read.

The project proposal is due at 5pm on Friday, November 18.

Final reports

The final report is due at 5 pm on Thursday, December 15. **This is a strict deadline.** Absolutely no extensions will be allowed. Any reports submitted after the deadline risk not being considered.

You may submit your project report electronically or on paper. If you submit the final report electronically, it must be in PDF format. If you submit on paper, place it in David Freeman's mailbox on the 4th floor of Gates; the mail room is Gates 465, across from David's office.

Format of the report

Your report should be typeset with LaTeX. If this is a serious hardship for you, come talk to me in advance.

Bibliography: Your bibliography entries should at a minimum have author(s), title, journal/book/conference title, and year. Volume number (for journals), publisher (for books) and page numbers would also be good. If the work is unpublished, give a url where it can be found.

Grading

The grade on the final project will be calculated as follows:

- 10%: Project proposal.

- 40%: Technical content.
- 25%: Mathematical correctness.
- 25%: Clarity of exposition. (This includes grammar and spelling as well as the organization of your paper. Make sure to proofread!)

Advice on writing

If you are not familiar with writing papers in mathematics or computer science (or even if you are), the following resources may help:

- Advice on research and writing from CMU:
  http://www.cs.cmu.edu/afs/cs.cmu.edu/user/mleone/web/how-to.html
- Henning Schulzrunni's advice on technical writing:
  http://www.cs.columbia.edu/~hgs/etc/writing-style.html
- Oded Goldreich's "How ~~not~~ to write a paper":
  http://www.wisdom.weizmann.ac.il/~oded/writing.html

This paper is an expository paper and is not expected to be original research. However, while the *ideas* in your paper do not need to be your own, I expect the *presentation* of these ideas to be your own work. Specifically, expository material such as introduction, background, discussion, and connecting sections should be written in your own words, and the overall viewpoint expressed in the paper (for example, why your topic is relevant and which technical contributions are important) should be your own. Copying such material from other sources without attribution will result in a failing grade on the paper.

Mathematical statements such as definitions, lemmas, and theorems in general *should* be quoted verbatim (with appropriate attribution), as their correctness depends on the precise use of terminology. Make sure, however, that your notation is consistent — if the same quantity is θ in one source and ω in another, you'll have to pick one.

When giving a proof it's fine to follow the original source closely, but I encourage filling in any gaps, giving additional explanation, and/or rephrasing some of the ideas. Basically I expect you to understand the concepts you are discussing and to be able to explain them in writing. When you copy and paste directly from others' work, I can't tell whether this understanding is there.

If you are following a particular source closely, you can say this at the beginning of the section. For example, "The following discussion is based on that of Author and Coauthor [4, Section 3]." Then you don't need to include a hundred citations in that section.

3 Topics

To whet your creativity, here are few possible ideas for projects. The projects described below are just a set of suggestions, and you may submit a proposal for a project based on any topic you like (not necessarily one based on a suggestion below). The only requirement is that the topic have something to do with elliptic curves in cryptography.

If you're looking for more ideas, a good place to start is the website of the annual Workshop on Elliptic Curve Cryptography workshop: http://eccworkshop.org. Find a talk that looks interesting and use Google

Scholar (http://scholar.google.com) to find the corresponding paper or papers. The author's website can also be useful. Don't limit yourself to recent work only!

The *Handbook of Elliptic and Hyperelliptic Curve Cryptography* (available online at http://www.crcnetbase.com/isbn/9781420034981) treats the mathematical side of ECC pretty comprehensively. You can skim the book to find a topic of interest and use the bibliography to find relevant research papers. (Such "bibliography diving" is a nice technique in general.) Pretty much any chapter of the Handbook could be the basis for a project.

Another place to look is the IACR eprint server: http://eprint.iacr.org. Search for papers with "elliptic" or "pairing" in the title, search by author, or just browse recent papers. (Beware that the quality of eprint submissions varies widely — make sure the paper you're using is correct and readable!)

Your topic must have some relevance to elliptic curve cryptography, but it need not be exclusively ECC. For example, analyzing discrete log algorithms for finite fields is relevant since it motivates the use of ECC; describing a cryptosystem that works in any finite group where discrete log is hard is relevant since it can be implemented using elliptic curves.

The topics below are loosely organized into categories; some topics may fit in more than one category.

Mathematics of elliptic and hyperelliptic curves

1. **Elliptic curve point counting.** Elkies and Atkin devised improvements to Schoof's algorithm that made elliptic curve point counting truly practical. Satoh and others devised methods to count points over fields of small characteristic.

   - Reynald, Lubicz, and Vercauteren, "Point counting on elliptic and hyperelliptic curves," Chapter 17 of *Handbook of Elliptic and Hyperelliptic Curve Cryptography.*
   - Schoof, "Elliptic curves over finite fields and the computation of square roots mod p," *Math. Comp.* 1985.
   - Elkies, "Elliptic and modular curves over finite fields and related computational issues," Computational Perspectives on Number Theory, 1988.
   - Satoh, "The canonical lift of an ordinary elliptic curve over a finite field and its point counting," *J. Ramanujan Math. Soc.* 2000.

2. **The complex multiplication method.** The theory of *complex multiplication* allows us to generate elliptic curves with known numbers of points.

   - Frey and Lange, "Complex multiplication," Chapter 18 of *Handbook of Elliptic and Hyperelliptic Curve Cryptography*
   - Bröker, "Constructing elliptic curves of prescribed order," Ph.D. Thesis, available at http://www.math.leidenuniv.nl/~reinier/thesis.pdf
   - Agashe, Lauter, and Venkatesan, "Constructing elliptic curves with a known number of points over a prime field," http://arxiv.org/abs/math/0111159

3. **Pairing-friendly elliptic curves.** Much work has gone into constructing ordinary elliptic curves for use in pairing-based cryptography. Such "pairing-friendly" curves have large prime-order subgroups and small embedding degree.

   - Miyaji, Nakabayashi, and Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Transactions on Fundamentals*, 2001.
   - Barreto and Naehrig, "Pairing-friendly elliptic curves of prime order," SAC 2006.

- Freeman, "Constructing pairing-friendly elliptic curves with embedding degree 10," ANTS 2006.
- Freeman, Scott, and Teske, "A taxonomy of pairing-friendly elliptic curves," *Journal of Cryptology* 2010.

4. **Relationship between DDH, CDH, and discrete log.** The best known algorithms to solve the decision Diffie-Hellman and computational Diffie-Hellman problems are to compute discrete logarithms. A major open question is whether one can do better.

   - Galbraith, "The Diffie-Hellman problem," http://www.math.auckland.ac.nz/~sgal018/crypto-book/ch22.pdf.
   - den Boer, "Diffie-Hellman is as strong as discrete log for certain primes," Crypto 1988
   - Maurer, "Maurer, Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms," CRYPTO 1994
   - Brown and Gallant, "The static Diffie-Hellman problem," http://eprint.iacr.org/2004/306.

5. **Generic groups.** A "generic algorithm" in a group is one that requires only the group operation and equality testing; it does not use any special structure of the group. Baby step-giant step is a generic algorithm, while index calculus in $\mathbb{F}_p^\times$ is non-generic (it uses the structure of representatives of integers mod p). Several authors have proposed a "generic group model" in which it can be proved, for example, that no generic discrete log algorithm in a group of size p has running time less than $\sqrt{p}$. Various extensions of the model allow for composite-order groups and groups with pairings.

   - Shoup, "Lower bounds for discrete logarithms and related problems," Eurocrypt 1997.
   - Boneh and Boyen, "Short signatures without random oracles and the SDH assumption in bilinear groups," J. Cryptology 2008.
   - Boyen, "The uber-assumption framework," Pairing 2008.

6. **Hyperelliptic curve point counting.** While in theory Schoof's algorithm generalizes to hyperelliptic curves, in practice the problem is much more difficult, and only recently have we been able to count points on curves of cryptographic size.

   - Reynald, Lubicz, and Vercauteren, "Point counting on elliptic and hyperelliptic curves," Chapter 17 of *Handbook of Elliptic and Hyperelliptic Curve Cryptography.*
   - Pila, "Frobenius maps of abelian varieties and finding roots of unity in finite fields," *Math. Comp.* 1990.
   - Gaudry and Harley, "Counting points on hyperelliptic curves over finite fields," ANTS 2000.
   - Gaudry and Schost, "Construction of secure random curves of genus 2 over prime fields," Eurocrypt 2004.
   - Gaudry and Schost, "Genus 2 point counting over prime fields," http://hal.inria.fr/inria-00542650.

7. **Index calculus on hyperelliptic curves.** There is no index calculus algorithm for elliptic curves, but on hyperelliptic curves there are index calculus algorithms. For fixed genus g these algorithms are still exponential in the group size, but for $g \geq 3$ they are faster than the generic (i.e., square-root) methods.

   - Avanzi and Thériault, "Index calculus for hyperelliptic curves," Chapter 21 of *Handbook of Elliptic and Hyperelliptic Curve Cryptography*
   - Adleman, DeMarrais, and Huang, "A subexponential algorithm for discrete logarithms over hyperelliptic curves of large genus over $GF(q)$", *Theoret. Comput. Sci.* 1999.
   - Gaudry, "An algorithm for solving the discrete log problem on hyperelliptic curves," Eurocrypt 2000.

- Enge and Gaudry, "A general framework for subexponential discrete logarithm algorithms," *Acta Arith.* 2002.
- Thériault, "Index Calculus Attack for Hyperelliptic Curves of Small Genus," Asiacrypt 2003.

Algorithms and implementation aspects

8. **Different coordinate systems.** In class we usually presented elliptic curves in Weierstrass form, $y^2 = x^3 + Ax + B$. However, in practice other coordinate systems, such as Jacobian coordinates or Edwards coordinates (see Washington Section 2.6) have performance advantages over Weierstrass coordinates. This project could have an implementation aspect: you could implement the various coordinate systems in SAGE and give some concrete timing data for comparisons.
   - Doche and Lange, Chapter 13 of *Handbook of Elliptic and Hyperelliptic Curve Cryptography.*
   - Cohen, Miyaji, and Ono, " Efficient elliptic curve exponentiation using mixed coordinates," Asicarypt 1998.
   - Bernstein, Lange, and Farashahi, "Binary Edwards curves," CHES 2008.
   - Bernstein, Birkner, Joye, Lange, Peters, "Twisted Edwards curves," Africacrypt 2008.
9. **Faster arithmetic using endomorphisms.** If an elliptic curve has an efficiently computable endomorphism, then point multiplication can be sped up significantly.
   - Gallant, Lambert, and Vanstone, "Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms," Crypto 2001.
   - Galbraith, Lin, and Scott, "Endomorphisms for faster elliptic curve cryptography on a large class of curves," Eurocrypt 2009.
10. **Hashing to elliptic curves.** Many crypto applications require hashing into elliptic curve groups; for example Boneh-Franklin IBE requires hashing identities to points. Efficient deterministic algorithms for this task have been discovered only recently.
    - Shallue and Woestijne, "Construction of Rational Points on Elliptic Curves over Finite Fields," ANTS 2006.
    - Icart, "How to Hash into Elliptic Curves," Crypto 2009.
    - Brier et al., "Efficient Indifferentiable Hashing into Ordinary Elliptic Curves," Crypto 2010.
    - Farashahi et al., "Indifferentiable Deterministic Hashing to Elliptic and Hyperelliptic Curves," preprint, http://eprint.iacr.org/2010/539.
11. **Speeding up pairing computation.** The basic algorithm for computing the Weil and Tate pairings is slow in practice. Many people have worked to improve these algorithms, sometimes defining new variants of the pairings.
    - Duquesne and Frey, "Implmentation of Pairings," Chapter 16 of *Handbook of Elliptic and Hyperelliptic Curve Cryptography.*
    - Barreto, Kim, Lynn, and Scott, "Fast algorithms for pairing-based cryptosystems," Crypto 2002.
    - Barreto, Galbraith, O'hÉigeartaigh, and Scott, "Efficient pairing computation on supersingular abelian varieties," *Designs, Codes, and Cryptography* 2007.
    - Scott, "Implementing cryptographic pairings," Pairing 2007.
    - Hess, Smart, and Vercauteren, "The eta pairing revisited," *IEEE Trans. Info. Theory* 2006.
    - Vercauteren, "Optimal pairings," *IEEE Trans. Info. Theory* 2010.

12. **Arithmetic of hyperelliptic curves.** The group law on hyperelliptic curves is much slower than that on elliptic curves, and much work has gone into optimizations.

    - Duquesne and Lange, "Arithmetic of Hyperelliptic Curves," Chapter 14 of *Handbook of Elliptic and Hyperelliptic Curve Cryptography*
    - Cantor, "Computing in the Jacobian of a hyperelliptic curve," *Math. Comp.* 1987.
    - Lange, "Efficient Arithmetic on Genus 2 Hyperelliptic Curves over Finite Fields via Explicit Formulae," http://eprint.iacr.org/2002/121.pdf.
    - Lauter, "The equivalence of the geometric and algebraic group laws for Jacobians of genus 2 curves," http://research.microsoft.com/en-us/um/people/klauter/ruthfig2.ps.

Attacks and defenses

13. **Discrete logs in finite fields.** The "number field sieve" is the best known method for computing discrete logarithms in finite fields of large characteristic. In characteristic 2 there is a slightly faster algorithm due to Coppersmith. Some of these algorithms have only heuristic running time analysis; one possibility for this project is to give a rigorous proof of the running time of the standard index calculus algorithm presented in class.

    - Galbraith, "Factoring and discrete logarithms in subexponential time," http://www.math.auckland.ac.nz/~sgal018/crypto-book/ch16.pdf
    - Odlyzko, "Discrete Logarithms: The Past and the Future," *Designs, Codes, and Cryptography* 2000.
    - Schirokauer, "Using number fields to compute logarithms in finite fields," *Math. Comp.* 2000.
    - Coppersmith, "Fast evaluation of logarithms in fields of characteristic two," *IEEE Trans. Inform. Theory* 1984.
    - Joux and Lercier, "The function field sieve in the medium prime case," Eurocrypt 2006.

14. **Speeding up Pollard rho.** Curve automorphisms can be used to improve the running time of Pollard's rho algorithm.

    - Duursma, Gaudry, and Morain, "Speeding up the discrete log computation on curves with automorphisms," Asiacrypt 1999.
    - Gallant, Lambert, and Vanstone, "Improving the parallelized Pollard lambda search on binary anomalous curves," *Mathematics of Computation* 2000.
    - Wiener and Zuccherato, "Faster attacks on elliptic curve cryptosystems," SAC 1998
    - Bernstein, Lange, and Schwabe, "On the correct use of the negation map in the Pollard rho method," PKC 2011.

15. **Side-channel attacks.** By measuring computation time, power consumed, and other "side-channel" information, it is possible to break elliptic curve cryptosystems without solving the underlying mathematical problem.

    - Byramjee, Courrège, and Feix, "Practical Attacks on Smart Cards," Chapter 28 of *Handbook on Elliptic and Hyperelliptic Curve Cryptography*
    - Kocher, "Timings attacks on implementations of Diffie-Hellman, RSA, DSS and other systems," Crypto 1996.
    - Kocher, Jaffe, Jun, "Differential power analysis," Crypto 1999.

- Biehl, Meyer, and Müller, "Differential fault attacks on elliptic curve cryptosystems," Crypto 2000.

16. **Countermeasures to side-channel attacks.** Since the discovery of side-channel attacks, researchers have worked to modify the algorithms so that less information is leaked — for example, so that processing a key bit of 0 takes the same amount of time as processing a key big of 1. Recently the theory community has introduced the notion of "leakage resilience" and provided provably secure constructions.

    - Lange, "Mathematical Countermeasures against Side-Channel Attacks," Chapter 29 of *Handbook of Elliptic and Hyperelliptic Curve Cryptography*
    - Avanzi, "Side Channel Attacks on Implementations of Curve-Based Cryptographic Primitives," http://eprint.iacr.org/2005/017
    - Faust, Kiltz, Pietrzak, and Rothblum, "Leakage-Resilient Signatures," TCC 2010.

Cryptosystems

17. **Elliptic curve systems mod N.** Several authors have proposed systems that use elliptic curves defined over $\mathbb{Z}_N$ where N is a large integer that is hard to factor. (Note that Paillier's proposal is broken.)

    - Koyama, Maurer, Okamoto, and Vanstone, "New Public-Key Schemes Based on Elliptic Curves over the Ring $\mathbb{Z}_n$," Crypto 1991.
    - Paillier, "Trapdooring Discrete Logarithms on Elliptic Curves over Rings," Asiacrypt 2000.
    - Galbraith, "Elliptic curve Paillier schemes," *J. Cryptology* 2002.
    - Galbraith, and McKee, "Pairings on elliptic curves over finite commutative rings," Cryptography and Coding 2005.
    - Dent and Galbraith, "Hidden pairings and trapdoor DDH groups," ANTS 2006.

18. **IBE schemes.** The Boneh-Franklin IBE scheme uses the random oracle model in its proof. Much work has gone into constructing schemes that do not require random oracles for their proofs of security.

    - Boneh and Boyen, "Efficient selective-ID secure identity based encryption without random oracles," Eurocrypt 2004.
    - Boneh and Boyen, "Secure identity based encryption without random oracles," Crypto 2004.
    - Waters, "Efficient identity-based encryption without random oracles," Eurocrypt 2005.
    - Gentry, "Practical identity-based encryption without random oracles," Eurocrypt 2006.
    - Waters, "Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions," Crypto 2009.

19. **Pairing-based signatures.** The BLS signature scheme provides basic signature functionality from the CDH assumption in the random oracle model. Other pairing-based signatures remove the random oracle and/or provide additional functionality.

    - Boneh and Boyen, "Short signatures without random oracles," *J. Cryptology 2008.*
    - Boneh, Gentry, Lynn, and Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," Eurocrypt 2003.
    - A. Boldyreva, "Efficient Threshold Signatures, Multisignatures and Blind Signatures based on the Gap-Diffie-Hellman-Group Signature Scheme," PKC 2003.

- Boyen and Waters, “Compact Group Signatures Without Random Oracles,” Eurocrypt 2006.
- Shacham and Waters, “Efficient Ring Signatures without Random Oracles,” PKC 2007.

20. **Hierarchical IBE.** In *hierarchical* identity-based encryption, the holder of a secret key can delegate secret keys to other users.
    - Horwitz and Lynn, “Toward hierarchical identity-based encryption,” Eurocrypt 2002.
    - Gentry and Silverberg, “Hierarchical ID-based cryptography,” Asiacrypt 2002.
    - Boneh, Boyen, and Goh, “Hierarchical identity-based encryption with constant size ciphertext,” Eurocrypt 2005.
    - Gentry and Halevi, “Hierarchical identity based encryption with polynomially many levels,” TCC 2009.
    - Waters, “Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions,” Crypto 2009.

21. **Functional encryption.** Another line of research that has developed from IBE is that of “functional encryption.” In a functional encryption scheme, a secret key is equipped with a list of properties and can decrypt any ciphertext that satisfies these properties.
    - Sahai and Waters, “Fuzzy identity-based encryption,” Eurocrypt 2005.
    - Goyal, Pandey, Sahai, and Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” ACM CCS 2006.
    - Bethencourt, Sahai, and Waters, “Ciphertext-policy attribute-based encryption,” IEEE Symposium on Security and Privacy 2007.
    - Boneh and Waters, “Conjunctive, subset, and range queries on encrypted data,” TCC 2007.
    - Katz, Sahai, and Waters, “Predicate encryption supporting disjunctions, polynomial equations, and inner products,” Eurocrypt 2008.