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 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 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 $\theta$ in one source and $\omega$ 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.

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

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.

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.

• 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

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 $\mathbb{F}_p^*$ 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.


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.
• Gaudry and Harley, “Counting points on hyperelliptic curves over finite fields,” ANTS 2000.
• Gaudry and Schost, “Genus 2 point counting over prime fields,” [http://hal.inria.fr/inria-00542650](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


### 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*.


9. **Faster arithmetic using endomorphisms.** If an elliptic curve has an efficiently computable endomorphism, then point multiplication can be sped up significantly.


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.


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*.


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*

**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.


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

- Wiener and Zuccherato, “Faster attacks on elliptic curve cryptosystems,” SAC 1998

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*

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 bit 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*

**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.)


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.


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.


20. **Hierarchical IBE.** In *hierarchical* identity-based encryption, the holder of a secret key can delegate secret keys to other users.


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.

- Goyal, Pandey, Sahai, and Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” ACM CCS 2006.