Formal proofs in applied cryptography

Daniel J. Bernstein

11 July 2023

Be prepared for a culture shock

Applied math is driven by service goals.
e.g. “Please speed up this computation.”

Be prepared for a culture shock

Applied math is driven by service goals.
e.g. “Please speed up this computation.”

Miscellaneous example, 1995 Bernstein:
Essentially-linear-time algorithm to detect perfect powers. Proof of time bound uses 1995 Loxton correction of 1986 Loxton “proof” in transcendental number theory.

Be prepared for a culture shock

Applied math is driven by service goals.
e.g. “Please speed up this computation.”

Miscellaneous example, 1995 Bernstein:
Essentially-linear-time algorithm to detect perfect powers. Proof of time bound uses 1995 Loxton correction of 1986 Loxton “proof” in transcendental number theory.

2004 Bernstein–Lenstra–Pila: Another such algorithm, simpler proof.

Be prepared for a culture shock

Applied math is driven by service goals.
e.g. “Please speed up this computation.”

Miscellaneous example, 1995 Bernstein:
Essentially-linear-time algorithm to detect perfect powers. Proof of time bound uses 1995 Loxton correction of 1986 Loxton “proof” in transcendental number theory.

2004 Bernstein–Lenstra–Pila: Another such algorithm, simpler proof. Applied view: Great, can skip the transcendental number theory!
Pure view: less proof depth; less interesting.

Do algorithms work correctly?

Most applications can simply test algorithms on typical examples and don't need higher assurance of algorithm correctness.

Do algorithms work correctly?

Most applications can simply test algorithms on typical examples and don't need higher assurance of algorithm correctness.

But high-risk applications care about proofs.
Maybe disastrous input randomly shows up.
Maybe attacker constructs disastrous input.

Do algorithms work correctly?

Most applications can simply test algorithms on typical examples and don’t need higher assurance of algorithm correctness.

But high-risk applications care about proofs. Maybe disastrous input randomly shows up. Maybe attacker constructs disastrous input.

Formal-proof example, 2023 Bernstein, for showing correctness of a recent fast modular-inversion algorithm: 3711-line Sage script producing 22771 lines in HOL Light proving the theorem on the next slide.

```
!i:num->int f:num->real g:num->real b:num m:num.
i(0) = &0 ==>
&0 <= g(0) ==>
g(0) <= f(0) ==>
f(0) <= &2 pow b ==>
(!n. (i(n+1) = &1 + i(n) /\ f(n+1) = f(n) /\ g(n+1) = g(n) / &2)
  \/ (if i(n) < &0
      then i(n+1) = &1 + i(n) /\ f(n+1) = f(n) /\ g(n+1) = (g(n)+f(n)) / &2
      else i(n+1) = - i(n) /\ f(n+1) = g(n) /\ g(n+1) = (g(n)-f(n)) / &2)
) ==>
(!n. integer(f(n))) ==>
(!n. integer(g(n))) ==>
9437 * b + 1 <= 4096 * m ==>
?n. n <= m /\ g(n) = &0
```

Exercise: Understand how proof uses the number $((1591853137+3\sqrt{273548757304312537})/2^{55})^{1/54}$.

My formalization goal this week

Theorem: Let n, t be nonnegative integers. Let k be a finite field with $\mathbf{F}_2 \subseteq k$. Let $\alpha_1, \ldots, \alpha_n$ be distinct elements of k. Define $A = \prod_i (x - \alpha_i)$. Let g be an element of $k[x]$ such that $\deg g = t$ and $\gcd\{g, A\} = 1$. Let B, a, b be elements of $k[x]$ with $\gcd\{a, b\} = 1$, $\deg a \leq t$, $A \in ak[x]$, and $\deg(aB - bA) < n - 2t + \deg a$. Assume that $g(\alpha_i)^2 B(\alpha_i)/A'(\alpha_i) \in \mathbf{F}_2$ for all i, where A' is the derivative of A. Define $e \in \mathbf{F}_2^n$ by $e_i = [a(\alpha_i) = 0]$. Then $\operatorname{wt} e = \deg a$ and

$$\sum_i \left(\frac{g(\alpha_i)^2 B(\alpha_i)}{A'(\alpha_i)} - e_i \right) \frac{A}{x - \alpha_i} \in g^2 k[x].$$

A correct-computation challenge

Among published post-quantum signature systems with no known feasible attacks: 2020 De Feo–Kohel–Leroux–Petit–Wesolowski "SQISign" has smallest total sig+key size.

Maybe this smallness will attract usage.

A correct-computation challenge

Among published post-quantum signature systems with no known feasible attacks: 2020 De Feo–Kohel–Leroux–Petit–Wesolowski “SQISign” has smallest total sig+key size.

Maybe this smallness will attract usage. (And that’s scary! Is SQISign secure?)

A correct-computation challenge

Among published post-quantum signature systems with no known feasible attacks: 2020 De Feo–Kohel–Leroux–Petit–Wesolowski "SQISign" has smallest total sig+key size.

Maybe this smallness will attract usage. (And that's scary! Is SQISign secure?)

Challenge: formally verify algorithms used for SQISign computation.

A correct-computation challenge

Among published post-quantum signature systems with no known feasible attacks: 2020 De Feo–Kohel–Leroux–Petit–Wesolowski "SQISign" has smallest total sig+key size.

Maybe this smallness will attract usage. (And that's scary! Is SQISign secure?)

Challenge: formally verify algorithms used for SQISign computation.

Concepts: supersingular elliptic curves over finite fields, isogenies, ideals of quaternion algebras, etc. See "Learning to SQI".

Are crypto constructions secure?

Crypto often advertises "security proofs":
type-*T* attack against construction *X*
with cost *c* and probability *p*
implies type-*U* attack against problem *Y*
with cost *d* and probability *q*.

Are crypto constructions secure?

Crypto often advertises "security proofs":
type-T attack against construction X
with cost c and probability p
implies type-U attack against problem Y
with cost d and probability q.

Common issues (see 2023 Koblitz–Menezes survey for many examples):

- Y isn't actually a hard problem.

Are crypto constructions secure?

Crypto often advertises "security proofs": type-T attack against construction X with cost c and probability p implies type-U attack against problem Y with cost d and probability q.

Common issues (see 2023 Koblitz–Menezes survey for many examples):

- Y isn't actually a hard problem.
- T is too narrow for the application.

Are crypto constructions secure?

Crypto often advertises "security proofs":
type-T attack against construction X
with cost c and probability p
implies type-U attack against problem Y
with cost d and probability q.

Common issues (see 2023 Koblitz–Menezes survey for many examples):

- Y isn't actually a hard problem.
- T is too narrow for the application.
- Big (c, p) vs. (d, q) gap is ignored.

Are crypto constructions secure?

Crypto often advertises “security proofs”:
type-T attack against construction X
with cost c and probability p
implies type-U attack against problem Y
with cost d and probability q.

Common issues (see 2023 Koblitz–Menezes survey for many examples):

- Y isn’t actually a hard problem.
- T is too narrow for the application.
- Big (c, p) vs. (d, q) gap is ignored.
- “Proof” is wrong.

A construction-security challenge

"Classic McEliece": a public-key encryption system using error-correcting codes.

Challenge: formalize the existing proof that any "QROM IND-CCA2" attack against Classic McEliece implies an "inversion" attack with comparable effectiveness against the original 1978 McEliece cryptosystem.

A construction-security challenge

"Classic McEliece": a public-key encryption system using error-correcting codes.

Challenge: formalize the existing proof that any "QROM IND-CCA2" attack against Classic McEliece implies an "inversion" attack with comparable effectiveness against the original 1978 McEliece cryptosystem.

Relies a bit on basic coding theory (using finite fields, matrices, polynomials) but main task is to formalize the proofs tracking cost and probability of quantum algorithms.
Warmup challenge: "ROM", non-quantum.

Are the underlying problems hard?

Very little change in McEliece inversion security levels since 1978 despite many papers attacking this inversion problem. We *hope* there isn't a much better attack.

Are the underlying problems hard?

Very little change in McEliece inversion security levels since 1978 despite many papers attacking this inversion problem. We *hope* there isn't a much better attack.

Most cryptographic problems are less well studied. Analogous hopes often fail.

Are the underlying problems hard?

Very little change in McEliece inversion security levels since 1978 despite many papers attacking this inversion problem. We *hope* there isn't a much better attack.

Most cryptographic problems are less well studied. Analogous hopes often fail.

General issue: We have no proofs of useful lower bounds on costs of high-Pr attacks.

Are the underlying problems hard?

Very little change in McEliece inversion security levels since 1978 despite many papers attacking this inversion problem. We *hope* there isn't a much better attack.

Most cryptographic problems are less well studied. Analogous hopes often fail.

General issue: We have no proofs of useful lower bounds on costs of high-Pr attacks.

And: Best proven performance among known attacks is much worse than best conjectured performance among known attacks.

Are proofs useless here?

Sometimes proofs for *components* of attack analyses help reduce risk of error.

e.g. 2023 Bernstein: 9950-line HOL Light proof of asymptotics of a particular function. Previous literature (1) uses this function as a model of the cost of lattice attacks and (2) makes claims about attack performance incompatible with these asymptotics.

Are proofs useless here?

Sometimes proofs for *components* of attack analyses help reduce risk of error.

e.g. 2023 Bernstein: 9950-line HOL Light proof of asymptotics of a particular function. Previous literature (1) uses this function as a model of the cost of lattice attacks and (2) makes claims about attack performance incompatible with these asymptotics.

Challenge: formally verify proofs given in 2021 Bernstein–Lange “Non-randomness of S-unit lattices”. Need cyclotomic fields, units, class groups, Brauer–Siegel theorem.