Algorithmic and mathematical methods for cryptology in France

G. Hanrot

INRIA Nancy Grand-Est – LORIA

MITACS, Dec. 7, 2007

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE
INRIA
centre de recherche NANCY - GRAND-EST

Outline

Context

- Mathematics mostly used in asymmetric (“public-key” cryptology) ; *Some underlying structure is needed to “hide” usable information in the public key* ;

Context

- Mathematics mostly used in asymmetric (“public-key” cryptology) ; *Some underlying structure is needed to “hide” usable information in the public key* ;
- Allows one to build *primitives* ;

Context

- Mathematics mostly used in asymmetric (“public-key” cryptology) ; *Some underlying structure is needed to “hide” usable information in the public key* ;
- Allows one to build *primitives* ;
- Some mathematical methods (lattices, Gröbner bases) are versatile tools in cryptanalysis ;

Context

- Mathematics mostly used in asymmetric (“public-key” cryptology) ; *Some underlying structure is needed to “hide” usable information in the public key* ;
- Allows one to build *primitives* ;
- Some mathematical methods (lattices, Gröbner bases) are versatile tools in cryptanalysis ;
- Two concerns on a given primitive :

Context

- Mathematics mostly used in asymmetric (“public-key” cryptology) ; *Some underlying structure is needed to “hide” usable information in the public key* ;
- Allows one to build *primitives* ;
- Some mathematical methods (lattices, Gröbner bases) are versatile tools in cryptanalysis ;
- Two concerns on a given primitive :
  - Evaluation of security (for primitives) to define suitable key-sizes, weak keys, etc.

Context

- Mathematics mostly used in asymmetric (“public-key” cryptology) ; *Some underlying structure is needed to “hide” usable information in the public key* ;
- Allows one to build *primitives* ;
- Some mathematical methods (lattices, Gröbner bases) are versatile tools in cryptanalysis ;
- Two concerns on a given primitive :
  - Evaluation of security (for primitives) to define suitable key-sizes, weak keys, etc.
  - Improve efficiency as much as possible.

Context

- Mathematics mostly used in asymmetric (“public-key” cryptology) ; *Some underlying structure is needed to “hide” usable information in the public key* ;
- Allows one to build *primitives* ;
- Some mathematical methods (lattices, Gröbner bases) are versatile tools in cryptanalysis ;
- Two concerns on a given primitive :
  - Evaluation of security (for primitives) to define suitable key-sizes, weak keys, etc.
  - Improve efficiency as much as possible.

Warning :

Context

- Mathematics mostly used in asymmetric (“public-key” cryptology) ; *Some underlying structure is needed to “hide” usable information in the public key* ;
- Allows one to build *primitives* ;
- Some mathematical methods (lattices, Gröbner bases) are versatile tools in cryptanalysis ;
- Two concerns on a given primitive :
  - Evaluation of security (for primitives) to define suitable key-sizes, weak keys, etc.
  - Improve efficiency as much as possible.

Warning :

- Primitives = *building blocks* ;

Context

- Mathematics mostly used in asymmetric (“public-key” cryptology) ; *Some underlying structure is needed to “hide” usable information in the public key* ;
- Allows one to build *primitives* ;
- Some mathematical methods (lattices, Gröbner bases) are versatile tools in cryptanalysis ;
- Two concerns on a given primitive :
  - Evaluation of security (for primitives) to define suitable key-sizes, weak keys, etc.
  - Improve efficiency as much as possible.

Warning :

- Primitives = *building blocks* ;
- MUST be combined with symmetric techniques to prevent access to underlying structure.

Number-based primitives

- Two underlying primitives : powering, exponentiation :

$$x \mapsto x^e \bmod N, n \mapsto g^n \bmod p \text{ or in a finite field.}$$

Number-based primitives

- Two underlying primitives : powering, exponentiation :

$$x \mapsto x^e \bmod N, n \mapsto g^n \bmod p \text{ or in a finite field.}$$

- RSA and El Gamal encryption/signature are built on top of this ;

Number-based primitives

- Two underlying primitives : powering, exponentiation :

$$x \mapsto x^e \bmod N, n \mapsto g^n \bmod p \text{ or in a finite field.}$$

- RSA and El Gamal encryption/signature are built on top of this ;
- Evaluating security $\Leftrightarrow$ try to invert those mappings ;

Number-based primitives (II)

Security

- Extract e-th roots or factor N (harder) in the first case ; best algorithm is NFS, complexity

$$L_N(1/3, (64/9)^{1/3}) := \exp((64/9 \log N (\log\log N)^2)^{1/3}).$$

Number-based primitives (II)

Security

- Extract e-th roots or factor N (harder) in the first case ; best algorithm is NFS, complexity

$$L_N(1/3, (64/9)^{1/3}) := \exp((64/9 \log N (\log\log N)^2)^{1/3}).$$

- Compute discrete logs in the second case ; best algorithms are NFS/FFS, complexity

$$L_q(1/3, O(1)).$$

Number-based primitives (II)

Security

- Extract e-th roots or factor N (harder) in the first case ; best algorithm is NFS, complexity

$$L_N(1/3, (64/9)^{1/3}) := \exp((64/9 \log N (\log\log N)^2)^{1/3}).$$

- Compute discrete logs in the second case ; best algorithms are NFS/FFS, complexity

$$L_q(1/3, O(1)).$$

Efficiency : fast finite field and ring arithmetic, efficient implementation in software or hardware.

Number-based primitives (II)

Security

- Extract e-th roots or factor N (harder) in the first case ; best algorithm is NFS, complexity

$$L_N(1/3, (64/9)^{1/3}) := \exp((64/9 \log N (\log\log N)^2)^{1/3}).$$

- Compute discrete logs in the second case ; best algorithms are NFS/FFS, complexity

$$L_q(1/3, O(1)).$$

Efficiency : fast finite field and ring arithmetic, efficient implementation in software or hardware.
Key-size : 2048 bits $\approx$ 600dd.

Number-based primitives (II)

Security

- Extract e-th roots or factor N (harder) in the first case ; best algorithm is NFS, complexity

$$L_N(1/3, (64/9)^{1/3}) := \exp((64/9 \log N (\log\log N)^2)^{1/3}).$$

- Compute discrete logs in the second case ; best algorithms are NFS/FFS, complexity

$$L_q(1/3, O(1)).$$

Efficiency : fast finite field and ring arithmetic, efficient implementation in software or hardware.
Key-size : 2048 bits $\approx$ 600dd.

Curves-based primitives

- Group structure on the set of points of a curve over a finite field (elliptic curves) or on a somewhat larger set (genus 2 curves) ;

Curves-based primitives

- Group structure on the set of points of a curve over a finite field (elliptic curves) or on a somewhat larger set (genus 2 curves) ;
- Two primitives $n \mapsto n \cdot P$; pairings $(P, Q) \mapsto e(P, Q)$ a bilinear form (allows for many advanced cryptographic constructions) ;

Curves-based primitives

- Group structure on the set of points of a curve over a finite field (elliptic curves) or on a somewhat larger set (genus 2 curves) ;
- Two primitives $n \mapsto n \cdot P$; pairings $(P, Q) \mapsto e(P, Q)$ a bilinear form (allows for many advanced cryptographic constructions) ;
- Evaluating security $\Leftrightarrow$ evaluate the difficulty of the discrete log problem ;

Curves-based primitives

- Group structure on the set of points of a curve over a finite field (elliptic curves) or on a somewhat larger set (genus 2 curves) ;
- Two primitives $n \mapsto n \cdot P$; pairings $(P, Q) \mapsto e(P, Q)$ a bilinear form (allows for many advanced cryptographic constructions) ;
- Evaluating security $\Leftrightarrow$ evaluate the difficulty of the discrete log problem ;
  - best general algorithm is Pollard-rho, complexity $\#G^{1/2}$;

Curves-based primitives

- Group structure on the set of points of a curve over a finite field (elliptic curves) or on a somewhat larger set (genus 2 curves) ;
- Two primitives $n \mapsto n \cdot P$; pairings $(P, Q) \mapsto e(P, Q)$ a bilinear form (allows for many advanced cryptographic constructions) ;
- Evaluating security $\Leftrightarrow$ evaluate the difficulty of the discrete log problem ;
  - best general algorithm is Pollard-rho, complexity $\#G^{1/2}$;
  - many weaker instances, starting to be well-known.

Curves-based primitives

- Group structure on the set of points of a curve over a finite field (elliptic curves) or on a somewhat larger set (genus 2 curves) ;
- Two primitives $n \mapsto n \cdot P$; pairings $(P, Q) \mapsto e(P, Q)$ a bilinear form (allows for many advanced cryptographic constructions) ;
- Evaluating security $\Leftrightarrow$ evaluate the difficulty of the discrete log problem ;
  - best general algorithm is Pollard-rho, complexity $\#G^{1/2}$;
  - many weaker instances, starting to be well-known.

Key-size : 192 bits $\approx$ 60dd.

Lattices and Gröbner bases

- Lattice = set of *integer* linear combinations of some linearly independent given vectors in $\mathbb{R}^n$;

Lattices and Gröbner bases

- Lattice = set of *integer* linear combinations of some linearly independent given vectors in $\mathbb{R}^n$;
- Gröbner basis = computationally convenient representation of a polynomial system ;

Lattices and Gröbner bases

- Lattice = set of *integer* linear combinations of some linearly independent given vectors in $\mathbb{R}^n$;
- Gröbner basis = computationally convenient representation of a polynomial system ;
- Important tools in cryptanalysis :

Lattices and Gröbner bases

- Lattice = set of *integer* linear combinations of some linearly independent given vectors in $\mathbb{R}^n$;
- Gröbner basis = computationally convenient representation of a polynomial system ;
- Important tools in cryptanalysis :
  - lattices are used to model and understand *linear phenomena*,

Lattices and Gröbner bases

- Lattice = set of *integer* linear combinations of some linearly independent given vectors in $\mathbb{R}^n$;
- Gröbner basis = computationally convenient representation of a polynomial system ;
- Important tools in cryptanalysis :
  - lattices are used to model and understand *linear phenomena*,
  - polynomial systems can model the behaviour of any boolean function.

Lattices and Gröbner bases

- Lattice = set of *integer* linear combinations of some linearly independent given vectors in $\mathbb{R}^n$;
- Gröbner basis = computationally convenient representation of a polynomial system ;
- Important tools in cryptanalysis :
  - lattices are used to model and understand *linear phenomena*,
  - polynomial systems can model the behaviour of any boolean function.
- Hard problems based on lattices (SVP, CVP) and on polynomial systems are used to build primitives.

A map of existing teams

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE
INRIA
centre de recherche NANCY - GRAND-EST

Topics addressed

- : teams partially supported by INRIA.
  - Factoring, primality proving : Nancy , Palaiseau ;
  - Discrete logs over finite fields : Rennes, Versailles ;
  - Algebraic cryptanalysis : Paris , Versailles ;
  - Algebraic curves for cryptology : Marseille, Nancy , Palaiseau , Rennes ;
  - Computer arithmetic, hardware for cryptology : Lyon , Montpellier, Nancy .
  - Lattices : Caen, Lyon , Paris .
  - Coding theory : Rocquencourt .

Zoom on the Nancy group (I)

- Permanent researchers : P. Gaudry, GH, E. Thomé, M. Videau, P. Zimmermann + 3 PhD students.

Zoom on the Nancy group (I)

- Permanent researchers : P. Gaudry, GH, E. Thomé, M. Videau, P. Zimmermann + 3 PhD students.
- Areas of expertise :

Zoom on the Nancy group (I)

- Permanent researchers : P. Gaudry, GH, E. Thomé, M. Videau, P. Zimmermann + 3 PhD students.
- Areas of expertise :
  - Algorithmic number theory (factoring, discrete logs, lattices) ;

Zoom on the Nancy group (I)

- Permanent researchers : P. Gaudry, GH, E. Thomé, M. Videau, P. Zimmermann + 3 PhD students.
- Areas of expertise :
  - Algorithmic number theory (factoring, discrete logs, lattices) ;
  - Algorithms for elliptic curves and jacobians of higher genus curves ;

Zoom on the Nancy group (I)

- Permanent researchers : P. Gaudry, GH, E. Thomé, M. Videau, P. Zimmermann + 3 PhD students.
- Areas of expertise :
  - Algorithmic number theory (factoring, discrete logs, lattices) ;
  - Algorithms for elliptic curves and jacobians of higher genus curves ;
  - Algorithms for fast and efficient arithmetic.

Zoom on the Nancy group (II)

Projects related to cryptology :

- Develop and optimize a complete implementation of the number field sieve, fully tuned in the whole range 100-150dd [joint project with Palaiseau, supported by French ANR] ;

Zoom on the Nancy group (II)

Projects related to cryptology :

- Develop and optimize a complete implementation of the number field sieve, fully tuned in the whole range 100-150dd [joint project with Palaiseau, supported by French ANR] ;
- Study curves of genus 2 and their applications in cryptology : efficient arithmetic, cardinality, discrete log ;

Zoom on the Nancy group (II)

Projects related to cryptology :

- Develop and optimize a complete implementation of the number field sieve, fully tuned in the whole range 100-150dd [joint project with Palaiseau, supported by French ANR] ;
- Study curves of genus 2 and their applications in cryptology : efficient arithmetic, cardinality, discrete log ;
- Speed up underlying arithmetics (finite field, p-adic) ;

Zoom on the Nancy group (II)

Projects related to cryptology :

- Develop and optimize a complete implementation of the number field sieve, fully tuned in the whole range 100-150dd [joint project with Palaiseau, supported by French ANR] ;
- Study curves of genus 2 and their applications in cryptology : efficient arithmetic, cardinality, discrete log ;
- Speed up underlying arithmetics (finite field, p-adic) ;
- Strong involvement in (efficient) software development.

Zoom on the Palaiseau group

- Permanent researchers : D. Augot (part-time), A. Enge, F. Morain, B. Smith + 1 PhD student.

Zoom on the Palaiseau group

- Permanent researchers : D. Augot (part-time), A. Enge, F. Morain, B. Smith + 1 PhD student.
- Areas of expertise :
  - Algorithmic number theory (factoring, primality proving) ;

Zoom on the Palaiseau group

- Permanent researchers : D. Augot (part-time), A. Enge, F. Morain, B. Smith + 1 PhD student.
- Areas of expertise :
  - Algorithmic number theory (factoring, primality proving) ;
  - Algorithms for elliptic curves, CM-type constructions ;

Zoom on the Palaiseau group

- Permanent researchers : D. Augot (part-time), A. Enge, F. Morain, B. Smith + 1 PhD student.
- Areas of expertise :
  - Algorithmic number theory (factoring, primality proving) ;
  - Algorithms for elliptic curves, CM-type constructions ;
  - Coding theory.

Zoom on the Palaiseau group

- Permanent researchers : D. Augot (part-time), A. Enge, F. Morain, B. Smith + 1 PhD student.
- Areas of expertise :
  - Algorithmic number theory (factoring, primality proving) ;
  - Algorithms for elliptic curves, CM-type constructions ;
  - Coding theory.
- Projects related to cryptology :
  - Factoring, see previous slide ;

Zoom on the Palaiseau group

- Permanent researchers : D. Augot (part-time), A. Enge, F. Morain, B. Smith + 1 PhD student.
- Areas of expertise :
  - Algorithmic number theory (factoring, primality proving) ;
  - Algorithms for elliptic curves, CM-type constructions ;
  - Coding theory.
- Projects related to cryptology :
  - Factoring, see previous slide ;
  - CM-type constructions ;

Zoom on the Palaiseau group

- Permanent researchers : D. Augot (part-time), A. Enge, F. Morain, B. Smith + 1 PhD student.
- Areas of expertise :
  - Algorithmic number theory (factoring, primality proving) ;
  - Algorithms for elliptic curves, CM-type constructions ;
  - Coding theory.
- Projects related to cryptology :
  - Factoring, see previous slide ;
  - CM-type constructions ;
  - Short elliptic signatures for adhoc networks ;

Zoom on the Palaiseau group

- Permanent researchers : D. Augot (part-time), A. Enge, F. Morain, B. Smith + 1 PhD student.
- Areas of expertise :
  - Algorithmic number theory (factoring, primality proving) ;
  - Algorithms for elliptic curves, CM-type constructions ;
  - Coding theory.
- Projects related to cryptology :
  - Factoring, see previous slide ;
  - CM-type constructions ;
  - Short elliptic signatures for adhoc networks ;
  - Applications of pairings to e-cash.

Fast arithmetic on jacobians of genus 2 curves

Background :

- Genus 2 curve = curve $y^2 = x^5 + \ldots$;

Fast arithmetic on jacobians of genus 2 curves

Background :

- Genus 2 curve = curve $y^2 = x^5 + \ldots$;
- Jacobian $\approx$ set of pairs of points on C ;

Fast arithmetic on jacobians of genus 2 curves

Background :

- Genus 2 curve = curve $y^2 = x^5 + \ldots$;
- Jacobian $\approx$ set of pairs of points on C ;
- Addition law : three pairs of points sum to 0 iff. there is a cubic curve $y = ax^3 + \ldots$ going through them ;

Fast arithmetic on jacobians of genus 2 curves

Background :

- Genus 2 curve = curve $y^2 = x^5 + \ldots$;
- Jacobian $\approx$ set of pairs of points on C ;
- Addition law : three pairs of points sum to 0 iff. there is a cubic curve $y = ax^3 + \ldots$ going through them ;
- Can be used as a cryptographic group over a finite field of size $\approx 2^{96}$ (“cheap” field arithmetic).

Fast arithmetic on jacobians of genus 2 curves

[P. Gaudry, J. Math. Cryptology 2007]
Goal : compute $n \cdot P$; basic operations = addition, doubling.
Count the number of such operations in the base field :

Fast arithmetic on jacobians of genus 2 curves

[P. Gaudry, J. Math. Cryptology 2007]
Goal : compute $n \cdot P$; basic operations = addition, doubling.
Count the number of such operations in the base field :

- Previous records : doubling = 34M+7S, addition = 37M+3S.
  On average $\log n$ doubling, $(\log n)/3$ additions.

Fast arithmetic on jacobians of genus 2 curves

[P. Gaudry, J. Math. Cryptology 2007]
Goal : compute $n \cdot P$; basic operations = addition, doubling.
Count the number of such operations in the base field :

- Previous records : doubling = 34M+7S, addition = 37M+3S.
  On average $\log n$ doubling, $(\log n)/3$ additions.
- New, very simple formulae based on θ functions :
  - 25M for one combined double/addition ;

Fast arithmetic on jacobians of genus 2 curves

[P. Gaudry, J. Math. Cryptology 2007]
Goal : compute $n \cdot P$; basic operations = addition, doubling.
Count the number of such operations in the base field :

- Previous records : doubling = 34M+7S, addition = 37M+3S.
  On average $\log n$ doubling, $(\log n)/3$ additions.
- New, very simple formulae based on θ functions :
  - 25M for one combined double/addition ;
  - Some are S ;

Fast arithmetic on jacobians of genus 2 curves

[P. Gaudry, J. Math. Cryptology 2007]
Goal : compute $n \cdot P$; basic operations = addition, doubling.
Count the number of such operations in the base field :

- Previous records : doubling = 34M+7S, addition = 37M+3S.
  On average $\log n$ doubling, $(\log n)/3$ additions.
- New, very simple formulae based on θ functions :
  - 25M for one combined double/addition ;
  - Some are S ;
  - Some are multiplications by a (possibly small) constant (possible speedup).

Fast arithmetic on jacobians of genus 2 curves

[P. Gaudry, J. Math. Cryptology 2007]
Goal : compute $n \cdot P$; basic operations = addition, doubling.
Count the number of such operations in the base field :

- Previous records : doubling = 34M+7S, addition = 37M+3S.
  On average $\log n$ doubling, $(\log n)/3$ additions.
- New, very simple formulae based on θ functions :
  - 25M for one combined double/addition ;
  - Some are S ;
  - Some are multiplications by a (possibly small) constant (possible speedup).

Practice : some implementations faster than elliptic curve for same security level (but pre-"Edwards coordinates").

Forging RSA signatures with affine padding

[A. Joux, D. Naccache, E. Thomé, *When e-th Roots Become Easier Than Factoring*, Asiacrypt' 2007].

- Some RSA-based signature schemes are of the form

$$S(m) = (h(m) + c)^d \bmod N = (h(m) + c)^{1/e} \bmod N,$$

where c is some padding constant. Assume $h = \mathrm{id}$ in the sequel.

Forging RSA signatures with affine padding

[A. Joux, D. Naccache, E. Thomé, *When e-th Roots Become Easier Than Factoring*, Asiacrypt' 2007].

- Some RSA-based signature schemes are of the form

$$S(m) = (h(m) + c)^d \bmod N = (h(m) + c)^{1/e} \bmod N,$$

  where c is some padding constant. Assume $h = \mathrm{id}$ in the sequel.
- For small padding, previous results show that forgery is possible in polynomial time ;

Forging RSA signatures with affine padding

For large padding (size $> 2/3 \log N$)

Forging RSA signatures with affine padding

For large padding (size $> 2/3 \log N$)

- Selective forgery is possible in time $L_N(1/3, (32/9)^{1/3})$, ie. faster than factoring ($L_N(1/3, (64/9)^{1/3})$).

Forging RSA signatures with affine padding

For large padding (size $> 2/3 \log N$)

- Selective forgery is possible in time $L_N(1/3, (32/9)^{1/3})$, ie. faster than factoring ($L_N(1/3, (64/9)^{1/3})$).

Practice : 512-bit selective forgery in a couple of hours on twenty machines.

Forging RSA signatures with affine padding

For large padding (size $> 2/3 \log N$)

- Selective forgery is possible in time $L_N(1/3, (32/9)^{1/3})$, ie. faster than factoring ($L_N(1/3, (64/9)^{1/3})$).

Practice : 512-bit selective forgery in a couple of hours on twenty machines.

Affine padding + RSA is weak.

Forging RSA signatures with affine padding

For large padding (size $> 2/3 \log N$)

- Selective forgery is possible in time $L_N(1/3, (32/9)^{1/3})$, ie. faster than factoring ($L_N(1/3, (64/9)^{1/3})$).

Practice : 512-bit selective forgery in a couple of hours on twenty machines.

Affine padding + RSA is weak.

Extra results : computing e-th roots faster than factoring given access to different type of oracles (and thus “general forgery”).