Primeless Modular Cryptography
(Extended Abstract)

Sonia Bogos, Ioana Boureanu, Serge Vaudenay

EPFL, Lausanne, Switzerland

Abstract. Most of the known public-key cryptosystems have an overall complexity which is dominated by the key-production algorithm, which requires the generation of prime numbers. This is most inconvenient in settings where the key-generation is not an one-off process, e.g., secure delegation of computation or EKE password-based key exchange protocols. To this end, we extend the Goldwasser-Micali (GM) cryptosystem to a provably secure system, denoted *SIS*, where the generation of primes is bypassed. Using number-theoretic and linear optimisation techniques, we align the security guarantees (i.e., resistance to factoring of moduli, etc.) of SIS to those of other well-known cryptosystems based on modular arithmetics. We explicitly compare and contrast the asymptotic complexity of well-known public-key cryptosystems based on modular arithmetics (e.g., GM and/or RSA) with that of SIS's. The latter shows that once we are ready to accept an increase in the size of the moduli, SIS's offers significant speed-ups to applications like the aforementioned secure delegation of computation or protocols where a fresh key needs to be generated with every new session. We also developed an efficient extension of SIS to handle more than one bit at a time, using linear codes, which will be omitted herein due to space constraints.

1 Introduction

Several, widely used public-key cryptosystems have a setup phase where prime numbers are generated and/or primality tests are run. The computational complexity yielded by the generation of a prime number of length L is generally in $\mathcal{O}(L^4)$ and –if optimised– $\mathcal{O}^{\sim}(L^3)$. Such generations occur, for instance, in the case of RSA [12] and/or in the Goldwasser-Micali (GM) probabilistic cryptosystem [3], as each of them defines its operation over $\mathbf{Z}_n^*$, for n being a product of two, distinct large prime numbers of size $L{=}s^3$ generated therein. Moreover, there exist settings in which the key-generation in asymmetric cryptosystems is not an one-off process. A first case of the sort is the more and more popular of *secure* delegation protocols [10], where some client outsources a task to a remote worker; the security of this is based on homomorphic public-key encryption schemes and the keys need to be re-issued freshly at each run of such a protocol. Hence, the asymptotic complexity of prime-generation for the homomorphic encryptions used therein [10] (e.g., GM, RSA, Paillier's encryption, encryption based on bilinear maps, etc.) becomes an alarming bottleneck of the delegated computation.

In this paper, we endeavor in extending the Goldwasser-Micali (GM) scheme [3] into a public-key scheme that bypasses prime-generation procedures. We show reduction in complexity from the usual $\mathcal{O}(s^{12})$ in the above cases to $\mathcal{O}(s^7(\log s)^2)$, at the cost of generating larger, composite numbers (where s is the security parameter). This comes to the special benefit of applications like the aforementioned (e.g., secure delegation of computation, EKE password-based key exchange protocols, etc.), where the key-generation giving the bottlenecks are in fact repeated at each run[1].

2 Preliminaries

2.1 Foundations

Let G be an Abelian group. A *character* χ is a group homomorphism from $(G,+)$ to $\mathbb{C}^*$. The set of characters over G has a group structure with component-wise multiplication over $\mathbb{C}^*$. For all $a \in G$, $\chi(a)$ is a $\lambda(G)$-th root of the unity, where $\lambda(G)$ is the exponent of the group G. A character χ of order 2 is such that $\chi(a) \in \{-1, 1\}$, for all $a \in G$. Let ε be the trivial character, i.e., $\varepsilon(a) = 1$. The set of characters χ for which $\chi^2{=}\varepsilon$ consists of ε and characters of order 2. Let $p \in \mathbb{Z}$ be an odd prime. The only character in $\mathbb{Z}_p^*$ of order 2 is the Legendre symbol $\chi(a) = (\frac{a}{p})$, for any $a \in \mathbf{Z}_p^*$. For $n{=}pq$ with p and q being two different odd primes, there are 3 characters of order 2: the Legendre symbol $(\frac{\cdot}{p})$, the Legendre symbol $(\frac{\cdot}{q})$, and the Jacobi symbol $(\frac{\cdot}{n})$. The latter is easy to compute, but the former are allegedly hard to compute when the primes p and q are unknown. We call these former characters *hard characters* of order 2. We recall that QR_n is a usual notation for the subgroup of Z_n^* of all quadratic residues. We refer to the problem of deciding whether an element of Z_n^* is quadratic residue or not as the QR problem.

This work uses characters of order 2, in order to design public-key encryption schemes that elude the generation of prime numbers, thus reducing the general asymptotic complexity of the usual schemes of the kind.

[1] We compare with (homomorphic) schemes used in these settings and do not compare with, e.g., the McEliece cryptosystem [9].

2.2 Computational Problems

In this paper, we consider the following combinatorial problem:

CHI Problem (Character Interpolation Problem):
Parameters: a modulus n, $x_1, \ldots, x_t$ in $\mathbf{Z}_n^*$, t elements $y_1, \ldots, y_t \in \{-1, +1\}$, all defining a unique character χ on $\mathbf{Z}_n^*$ such that $\chi(x_i) = y_i$ for $i = 1, \ldots, t$ and $t \geq 1$.
Input: $x \in \mathbf{Z}_n^*$.
Problem: Find $y = \chi(x)$.

CHI is a specialisation of the **GHI** problem [11]. When one can compute discrete logarithms in $\mathbf{Z}_n^*$ or factor n, one can easily solve the **CHI** problem by solving a linear system. Thus, for the **CHI** problem to be hard, we need that n is hard-to-factor. The hardness of the **CHI** and the **QR** problems are formally defined as expected, i.e., in negligible advantages of ppt. adversaries trying to defeat the assumptions. Then, we have the following (amplification) result:

Theorem 1. *If the QR problem is hard relative to Gen_{GM}, then **CHI** problem is hard relative to Gen_{CHI}.*

We assume that **the best algorithm to solve CHI problem with $\chi(\cdot) = \left(\frac{\cdot}{\alpha}\right)$ over $\mathbb{Z}_n^*$, for a factor α of n, consists of finding α.** Our main cryptosystem then relies on this problem.

3 `SIS`: A Primeless Public-Key Cryptosystem

`SIS` **Modulus generation:**
Input: Security parameter s.
1: compute k and ℓ (depending on s, using in (9) and (8) in page 4)
2: pick $\alpha = \prod_{i=1}^{i=k} \alpha_i$, where α_i are random odd integers of size ℓ;
3: pick $\beta = \prod_{i=1}^{i=k} \beta_i$, where β_i are random odd integers of size ℓ;
4: compute $n = \alpha \cdot \beta$
Output: Public key: (n, α).

`SIS Key generation:`
Input: Security parameter s.
1: compute t (depending on s, as per (2) in page 3)
2: $(n, \alpha) \leftarrow$ `GenModulus`(s)
3: $x_1, x_2, \ldots, x_t \in_U \mathbf{Z}_n^*$
4: $y_i = \left(\frac{x_i}{\alpha}\right)$ for all $1 \leq i \leq t$
5: **if** y_i=1 for all $1 \leq i \leq t$, **then** go-to step 4
Output: Public key: $(n, x_1, x_2, \ldots x_t, y_1, y_2, \ldots y_t)$; Private key: α.

`SIS Encryption:`
Input: a bit b.
Public key: $(n, x_1, x_2, \ldots x_t, y_1, y_2, \ldots y_t)$.
1: find y_i=-1, $i \in \{1, \ldots, t\}$
2: $b_1, b_2, \ldots, b_{i-1}, b_{i+1}, \ldots, b_t \in_U \{0, 1\}$
3: $P \leftarrow \prod_{j \neq i} y_j^{b_j}$
4: **if** P=$(-1)^b$ **then** $b_i \leftarrow 0$ **else** $b_i \leftarrow 1$.
5: $z' \leftarrow x_1^{b_1} \cdots x_t^{b_t} \pmod n$
6: $r \in_U \mathbf{Z}_n^*$
7: $z \leftarrow r^2 \cdot z' \pmod n$
Output: the encryption z, $z \in \mathbf{Z}_n^*$.

`SIS Decryption:`
Input: the encryption z, $z \in \mathbb{Z}_n^*$.
Secret key: α.
1: Compute the Jacobi symbol $\left(\frac{z}{\alpha}\right)$.
2: **if** $\left(\frac{z}{\alpha}\right) = 1$ **then** $b = 0$ **else** $b = 1$.
Output: a bit b.

The cryptosystem is presented in the above figure. The security implied by procedures 1-4 and the size of their parameters are discussed in Section 4.2. Also, the system is correct, i.e.,:

Lemma 2. $(\frac{z}{\alpha}) = (-1)^b$, *where the values* z, α, *the bit* b *are honestly computed/selected as in the* SIS *cryptosystem.*

4 Description & Choice of the Parameters

4.1 The Local Parameter t

Let $s \in \mathbb{Z}$ be the security parameter and L be the bitlength of n. The value L is given in terms of s is expressed at the end of this section. We pick the value t such that we obtain the uniqueness of the homomorphism in the **GHI** corresponding problem. Namely, we pick t to be greater than the value r specified by Lemma 4.3 in [11], specialised here for d=2. When $\{x_1, \ldots x_t\}$ is such that no different characters collide on this set, we say that $\mathbb{Z}_2$-generates $\mathbb{Z}_n^*$. Using Theorem 4.29 in [11], we give the following corollary:

Lemma 3. *The probability that* $\{x_1, \ldots, x_t\}$ *in the* SIS *scheme* $\mathbf{Z}_2$*-generate* $\mathbf{Z}_n^*$ *is* $P_{gen} \geq 1 - 2^{k_2 - t}$, *where* k_2 *is the rank of the group* A_2 *and* A_2 *is the maximal* 2*-subgroup of* $\mathbf{Z}_n^*$.

In order to enforce that $1 - P_{gen}$ is smaller than 2^{-s}, we get a sufficient bound for t: $t \geq k_2 + s$. Further, the rank k_2 of the 2-subgroup of $\mathbf{Z}_n^*$ is closely related to $\omega(n)$, i.e., the number of distinct prime factors of n [4]. Since n to be generated in the SIS scheme is odd, we can conclude the number t of elements used from $\mathbf{Z}_n^*$ to generate z' is such that $t \geq \omega(n) + s$. This is a sufficient condition for $P_{\mathsf{gen}} \geq 1 - 2^{-s}$.

Theorem 4. *For* $t \geq \omega(n) + s$, $x_1, \ldots, x_t$ $\mathbb{Z}_2$*-generate* $\mathbb{Z}_n^*$ *with a probability greater than or equal to* $1 - 2^{-s}$.

By using he Ramanujam-Hardy theorem [4], the Erdös-Kac theorem [2] and the approximation of the standard normal cumulative distribution, we obtain that (2) $t = \lceil 2k. \ln\ln 2^\ell + \sqrt{2s. \ln 2. \ln\ln 2^\ell} + s \rceil$. Hence, t can be taken of the order of $k \ln \ell + s$.

4.2 Asymptotic Approximations & Numerical Optimisation of k, ℓ, t

It can be seen that in order for the **CHI** problem to be hard and, separately for key recovery attacks to be impossible, the factorization of the modulus n, generated as in our cryptosystem, needs to be hard. More precisely, the factors α and β of n should be hard to find.

In [7], Knuth *et al.* look at on the probability that, for a random number n, the r^{th} largest of its prime factors, n_r, is smaller than n^x where $0 < x < 1$: $F_r(x) = \lim_{N \to +\infty} \frac{P_r(x,N)}{N}$, where $P_r(x, N)$ is the following function $P_r(x, N) = \#\{1 \leq n \leq N | n_r \leq N^x\}$. We use this to express our security desiderata (the hardness on n's factorization); Let some constant $x, y, z \in (0, 1)$ and consider the following. Let $C_{GNFS}(|n|, c, \varepsilon){=}c \times (e^{\left(\sqrt[3]{\frac{64}{9}}+\varepsilon\right)(\ln n)^{\frac{1}{3}}(\ln\ln n)^{\frac{2}{3}}}$ and $C_{ECM}(|p|, c', \varepsilon'){=}c' \times e^{\sqrt{2+\varepsilon'}\sqrt{\ln p \ln\ln p}}$ express the complexity of factorizing n using the GNFS (general number field sieve) and ECM (elliptic curve method) methods respectively, where p is the smallest prime factor of n, $c, c' \in \mathbb{R}$ is such that $C_{GNFS}(1248){=}2^{80}$ [1]. Now, we impose our conditions to align the security of SIS to the security levels of factorising moduli in general, in public-key cryptography:

(3) $\min\left[F_1(x)^{-k}, \min_{x \leq u \leq y} \frac{C_{ECM}(u \cdot \ell)}{F_1(u)^k}, C_{ECM}(y \cdot \ell)\right] \geq 2^s$; (4) $\Pr\left[\#\{i : \alpha_i \text{ is } 2^{y \cdot \ell}\text{-smooth}\} > zk\right] \leq 2^{-s}$; (5) $F_1(y) \leq z$; (6) $exp(-2k(F_1(y) - z)^2) \leq 2^{-s}$; (7) $C_{GNFS}((1 - z)ky\ell) \geq 2^s$. Let s be the security parameter. Equation (3) stipulates that, for all fractions u between a fraction x (of our numbers α_i) and a fraction y, either the complexity to find factors of size $u\ell$ in α generated as above is too high or the probability that all α_i are $2^{u\ell}$-smooth is too low. Then, to select our bitlength $L = 2k\ell$ aligned to that of moduli used inside schemes like that of Goldwasser-Micali [3] or inside RSA [12] and to equate their security guarantees, we consider that the following needs to hold: $C_{GNFS}(L) \geq 2^s$. From this, we approximate that $L = \mathcal{O}^{\sim}(s^3)$. If instead we consider that ECM [8] is used to factorize moduli, then we approximate that ℓ is of the order of s^2. We should also ensure that the number of hard-to-find factors in α is high and larger than $2^{y \cdot \ell}$, i.e., for their product to resist factoring with GNFS. Hence, our requirement (4), where z is a constant, α_i are as in the GenModulus algorithm. So, we have $(1 - z)k$ prime factors of size larger than $y\ell$. (We also use this information to construct a cryptosystem that encrypts more than 1 bit.)

From the de Brujin function ψ [5], we can approximate that the number of $2^{y\ell}$-smooth factors smaller than zk, as needed above. We now use the Hoeffding inequality [6] upon requirement (4), for the case where (5) is the case. Then, the probability in (4) is lower than $exp(-2k(F_1(y) - z)^2)$. Overall we require that this probability is smaller than 2^{-s}, i.e., (6).

We need to find the bounds of L (or ℓ) imposed by our criterion (4) above. We consider that x and y are constant. By the constraints on k following from above (i.e., $exp(-2k(F_1(y)-z)^2) \leq 2^{-s}$ and $F_1(x)^{-k} \leq 2^{-s}$), then z can be chosen constant. Indeed, for (4) to hold, we enforce at the lower bound that $C_{GNFS}((1-z)ky\ell)$ is at least equal to 2^s, i.e., (7). Taking x=y, z being fixed, we let $k \in \mathcal{O}(s)$ satisfying (6) and (3). Then, $\ell \in \mathcal{O}(s^2)$ satisfies (3) and (7). Since we already showed that $t \in \mathcal{O}(k \ln \ell + s)$, we have the following overall result:

t	ℓ	k	$k\ell$
$\mathcal{O}(s \log s)$	$\mathcal{O}(s^2)$	$\mathcal{O}(s)$	$\mathcal{O}(s^3)$

Now, we mention some of the series of values obtained using all the formulae in the previous section and/or others derived from it, e.g., (8) $\ell = \frac{C^{-1}_{ECM}(2^s)}{x}$, (9) $k = \frac{s \ln 2}{(z-F_1(x))^2}$, implemented in PARI/GP, using an approximation of the DeBrujin/Dickman's functions as per [7]. Thus, for s=80, we obtained $x = y$=0.470, z=0.961, k=57, ℓ=841, $k\ell$=47 937, t=464. For $s = 192$, $k\ell$=503 712.

5 Complexity & Security of the Scheme

Complexity. We evaluated the complexity of our schemes, compared it with that of other public-key cryptosystems based on primality-testing, e.g., GM and RSA. We report a small part of these evaluations, which show that `SIS` exhibits improved asymptotic complexities for all procedures, apart from encryption:

		key-generation	encryption	decryption
schoolbook multiplication	GM	$\mathcal{O}(s^{12})$	$\mathcal{O}(s^6)$	$\mathcal{O}(s^6 \log s)$
	SIS	$\mathcal{O}(s^7(\log s)^2)$	$\mathcal{O}(s^7 \log s)$	$\mathcal{O}(s^6 \log s)$
FFT-based multiplication	GM	$\mathcal{O}(s^9 \log s)$	$\mathcal{O}(s^3(\log s))$	$\mathcal{O}(s^3(\log s)^2)$
	SIS	$\mathcal{O}(s^4(\log s)^3)$	$\mathcal{O}(s^4(\log s)^2)$	$\mathcal{O}(s^3(\log s)^2)$

Security.

Theorem 5. *Assuming that the **CHI** problem is hard relative to Gen, the `SIS` scheme is IND-CPA secure.*

Since the `SIS`-scheme is homomorphic, it is clearly not IND-CCA secure.

6 Conclusions

Relying on hard characters of order 2, we have extended the GM cryptosystem in a way that bypasses completely the use/generation of prime numbers. In doing so, the resulting scheme yields an asymptotic complexity in the security parameter smaller than the one of standard public-key cryptosystems. This would yield a considerable speed-up to secure delegation protocols [10] that use homomorphic encryption schemes, GM included, in a way where the key-generation is repeated at every run of the protocol. In the extended paper, we show how encrypt more than 1 in an efficient way, using linear codes. It is also possible to improve the efficiency of our cryptosystem by using characters of higher order.

References

1. Institute of Electrical and Electronics Engineers. ECRYPT II Yearly Report on Algorithms and Keysizes. ECRYPT, 2011. `http://www.ecrypt.eu.org/documents/D.SPA.17.pdf`.
2. P. Erdös and M. Kac. The gaussian law of errors in the theory of additive number theoretic functions. *American Journal of Mathematics*, 62(1):738–742, 1940.
3. S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.
4. G. Hardy and S. Ramanujan. The normal number of prime factors of a number *n*. *Quart. J. Math.*
5. A. Hildebrand and G. Tenenbaum. *Integers without large prime factors.* Prépublications de l'Institut Elie Cartan. Dép. de Math., Univ. de Nancy I, 1991.
6. W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.
7. D. E. Knuth and L. T. Pardo. Analysis of a simple factorization algorithm. *Theoretical Computer Science*, 3(3):321348, 1976.
8. H. W. Lenstra, Jr. Factoring integers with elliptic curves. *Ann. of Math. (2)*, 126(3):649–673, 1987.
9. Robert J. Mceliece. A public-key cryptosystem based on algebraic coding theory. Technical report, Jet Propulsion Lab Deep Space Network Progress report, 1978.
10. P. Mohassel. Efficient and secure delegation of linear algebra. Cryptology ePrint Archive, Report 2011/605, 2011. `http://eprint.iacr.org/`.
11. J. Monnerat and S. Vaudenay. Short Undeniable Signatures Based on Group Homomorphisms. *Journal of Cryptology*, 24(3):545–587, 2011.
12. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21:120–126, February 1978.