Published in J.A. Garay, A. Miyaji, and A. Otsuka, Eds, *Cryptology and Network Security (CANS 2009)*, vol. 5888 of *Lecture Notes in Computer Science*, pp. 41–52, Springer, 2009.

On Cryptographic Schemes Based on Discrete Logarithms and Factoring

Marc Joye

Thomson R&D, Security Competence Center
1 avenue de Belle Fontaine, 35576 Cesson-Sévigné Cedex, France
marc.joye@thomson.net – http://joye.site88.net/

Abstract. At CRYPTO 2003, Rubin and Silverberg introduced the concept of torus-based cryptography over a finite field. We extend their setting to the ring of integers modulo N. We so obtain compact representations for cryptographic systems that base their security on the discrete logarithm problem and the factoring problem. This results in smaller key sizes and substantial savings in memory and bandwidth. But unlike the case of finite fields, analogous trace-based compression methods cannot be adapted to accommodate our extended setting when the underlying systems require more than a mere exponentiation. As an application, we present an improved, torus-based implementation of the ACJT group signature scheme.

Keywords. Torus-based cryptography, ring $\mathbb{Z}_N$, discrete logarithm problem, factoring problem, compression, ACJT group signatures.

— *To Isabelle Déchène*

1 Introduction

Groups where the discrete logarithm problem is assumed to be intractable are central in the design of public-key cryptography. This was first pointed out by Diffie and Hellman in their seminal paper [7]. The security of the Diffie-Hellman key-distribution system relies on the intractability of the discrete logarithm problem in the multiplicative group of finite fields. Such groups also allow one to construct encryption schemes, digital signature schemes, and many other cryptographic primitives and protocols [19].

Several improvements were proposed to improve the efficiency of the so-obtained schemes. In [24], Schnorr suggests to work in a prime-order subgroup of $\mathbb{F}_p^\times$ rather than in the whole group $\mathbb{F}_p^\times$. Building on [26], Lenstra [15] extends this idea to the cyclotomic subgroup of $\mathbb{F}_{p^r}^\times$. He states that the underlying field is really $\mathbb{F}_{p^r}$ and not some intermediate subfield thereof. More recently, Rubin and Silverberg [21] (see also [22, 23]) rephrased cyclotomic subgroups in terms of algebraic tori over $\mathbb{F}_p$. They also proposed the CEILIDH cryptosystem. The main advantage of their approach resides in the compact representation of the

elements. Previous prominent proposals featuring a compact representation include LUC [26] and XTR [16]. Several speedups and simplifications for XTR are described in [28]. Optimizations for CEILIDH and a comparison with XTR can be found in [11].

Variants of Diffie-Hellman key-distribution system in the multiplicative group $\mathbb{Z}_N^\times$ where N is the product of two primes are proposed in [17, 25]. The goal is to combine the security of the original scheme with the difficulty of factoring large numbers. In [17], McCurley argues that it may be desirable to design cryptographic systems with the property that breaking them requires to solve two different computational problems.

In this paper, we introduce torus-based cryptography over the ring $\mathbb{Z}_N$. It finds applications in settings similar to those considered by McCurley. It also reveals useful to increase the performance of cryptographic schemes whose security requires *both* the integer factorization assumption and the discrete logarithm assumption, or related assumptions (e.g., [3, 2, 10]). Substantial savings both in memory and in transmission are achieved without security loss. The representation used in [27] offers the same savings as one-dimensional tori over $\mathbb{Z}_N$. Unfortunately, its usage is mostly restricted to exponentiation: [27] presents an analogue of RSA. Numerous applications however require full use of multiplication. Tori over $\mathbb{Z}_N$ embed a group structure and therefore suit a much wider range of applications. We consider this as the main feature of torus-based cryptography.

As an illustration, we consider the ACJT group signature scheme [1], used in the design of the protocol standardized by the Trusted Computing Group [29] to protect privacy of the device's user. Group signature schemes, as introduced by Chaum and van Heyst [5], allow a group member to sign anonymously on behalf of the group. However, the group manager is able to recover the signer's identity. The ACJT scheme makes use of arithmetic modulo N, where $N = pq$ is a strong RSA modulus. Each group member possesses a membership certificate $[A, e]$ satisfying $A^e = a^x\, a_0 \pmod N$ where $\{a, a_0, N\}$ are common public parameters and x denotes the member's private key. *As the group manager may know the factorization of N, the secrecy of private key x is only guaranteed modulo p and q. As remarked in [4], if we wish to disallow the group manager to frame group members, the length of modulus N should typically be doubled.* Based on current understanding, a torus-based implementation offers the same security level but without requiring to increase the length of N. For example, for an expected 80-bit security level, the size of the resulting signatures is about 11 kb (this is half the amount of the original scheme) and the generation of a signature is more than three times faster.

The rest of this paper is organized as follows. In the next section, we provide some background on algebraic tori. We present a compact representation of one-dimensional tori from the geometric interpretation of the group law on Pell conics. We also mention compact representations for higher-dimensional tori. In Section 3, we extend torus-based representations over rings. The main focus is

put on the ring $\mathbb{Z}_N$ where N is an RSA modulus. We compare the so-obtained representations with Lucas-based representations and explain why the latter are not appropriate. Section 4 addresses applications of our torus-based compression. We present a detailed implementation of the ACJT group signature scheme using a torus-based representation and discuss the performance of the resulting scheme. Finally, we conclude in Section 5.

2 Torus-Based Cryptography

Let $\mathbb{F}_q$ denote the finite field with $q = p^r$ elements. The order of the multiplicative group $\mathbb{F}_{p^r}^\times = \mathbb{F}_{p^r} \setminus \{0\}$ is $p^r - 1$. Note that $p^r - 1 = \prod_{d|r} \Phi_d(p)$ where $\Phi_d(x)$ represents the r-th cyclotomic polynomial. We let $\mathsf{G}_{p,r} \subseteq \mathbb{F}_{p^r}^\times$ denote the cyclic subgroup of order $\Phi_r(p)$.

In [21], Rubin and Silverberg identify $\mathsf{G}_{p,r}$ with the $\mathbb{F}_p$-points of an algebraic torus $\mathcal{T}_r(\mathbb{F}_p)$. Namely, they consider

$$\mathcal{T}_r(\mathbb{F}_p) = \{\alpha \in \mathbb{F}_{p^r}^\times \mid N_{\mathbb{F}_{p^r}/F}(\alpha) = 1 \text{ whenever } \mathbb{F}_p \subseteq F \subsetneq \mathbb{F}_{p^r}\},$$

that is, the elements of $\mathbb{F}_{p^r}^\times$ whose norm is one down to every intermediate subfield F. The key observation is that $\mathcal{T}_r(\mathbb{F}_p)$ forms a group whose elements can be represented with only $\varphi(r)$ elements of $\mathbb{F}_p$, where φ denotes Euler's totient function. The compression factor is thus of $r/\varphi(r)$ over the field representation [21, 9].

2.1 Parametrization of $\mathcal{T}_2(\mathbb{F}_p)$

We detail a compact representation of $\mathcal{T}_r(\mathbb{F}_p)$ for the case $r = 2$. We have $|\mathbb{F}_{p^2}^\times| = p^2 - 1$, $\Phi_2(p) = p + 1$, and $\mathsf{G}_{p,2} = \{\alpha \in \mathbb{F}_{p^2}^\times \mid \alpha^{\Phi_2(p)} = 1\}$. We assume p odd and write $\mathbb{F}_{p^2} = \mathbb{F}_p(\sqrt{\Delta})$ for some non-square $\Delta \in \mathbb{F}_p^\times$. We have $\mathsf{G}_{p,2} = \{x + y\sqrt{\Delta} \mid x, y \in \mathbb{F}_p \text{ and } (x + y\sqrt{\Delta})^{p+1} = 1\}$. Since $(x + y\sqrt{\Delta})^p = x - y\sqrt{\Delta}$, it follows that $(x + y\sqrt{\Delta})^{p+1} = (x - y\sqrt{\Delta})(x + y\sqrt{\Delta}) = x^2 - \Delta y^2$.

So, the group $\mathsf{G}_{p,2}$ can be seen as the set of $\mathbb{F}_p$ points on the genus 0 curve C over $\mathbb{F}_p$ given by the Pell equation

$$C_{/\mathbb{F}_p} : x^2 - \Delta\, y^2 = 1 \ . \tag{1}$$

We have $\mathsf{G}_{p,2} \cong \mathcal{T}_2(\mathbb{F}_p) \cong C(\mathbb{F}_p)$ [21, Lemma 7] (see also [18, Theorem 4.5]). If we denote by $\oplus$ the group law on $C(\mathbb{F}_p)$, given two points $(x_1, y_1), (x_2, y_2) \in C(\mathbb{F}_p)$, we have

$$(x_1, y_1) \oplus (x_2, y_2) = (x_1x_2 + \Delta\, y_1y_2, x_1y_2 + x_2y_1) \ .$$

The neutral element is $\boldsymbol{\mathcal{O}} = (1, 0)$ and the inverse of (x, y) is $(x, -y)$.

As remarked in [6, Chapter 3], the geometric interpretation of the group law on $C(\mathbb{F}_p)$ gives rise to a compact representation. Let $\boldsymbol{P} = (x_1, y_1)$ and

$\boldsymbol{Q} = (x_2, y_2)$ be two points of $C(\mathbb{F}_p)$. The group law on $C(\mathbb{F}_p)$ is given by the so-called 'chord-and-tangent' rule [20] (see also [14, § 1] for a detailed account). We denote by $\ell_{\boldsymbol{P},\boldsymbol{Q}}$ the line passing through $\boldsymbol{P}$ and $\boldsymbol{Q}$; $\ell_{\boldsymbol{P},\boldsymbol{Q}}$ represents the tangent line at $\boldsymbol{P}$ if $\boldsymbol{P} = \boldsymbol{Q}$. The parallel line, say ℓ', to $\ell_{\boldsymbol{P},\boldsymbol{Q}}$ that passes through $\boldsymbol{\mathcal{O}} = (1, 0)$ intersects (counting multiplicity) the Pell conic C at precisely one other point (x_3, y_3), which is defined as $\boldsymbol{P} \oplus \boldsymbol{Q}$. If m denotes the slope of $\ell_{\boldsymbol{P},\boldsymbol{Q}}$ then the equation of ℓ' is given by $y = m(x - 1)$. Therefore, (x_3, y_3) satisfies ${x_3}^2 - \Delta\,{y_3}^2 = 1$ and $y_3 = m(x_3 - 1)$. We get ${x_3}^2 - \Delta\, m^2(x_3 - 1)^2 = 1 \iff (x_3 - 1)\big(x_3(1 - \Delta\, m^2) + \Delta\, m^2 + 1\big) = 0$. From $y_3 = m(x_3 - 1)$, we find

$$(x_3, y_3) = \left(\frac{\Delta\, m^2 + 1}{\Delta\, m^2 - 1}, \frac{2m}{\Delta\, m^2 - 1}\right) = \left(\frac{(\Delta\, m)^2 + \Delta}{(\Delta\, m)^2 - \Delta}, \frac{2(\Delta\, m)}{(\Delta\, m)^2 - \Delta}\right) \ .$$

Let now $\boldsymbol{P} = (x, y)$ be a point in $C(\mathbb{F}_p) \setminus \{\boldsymbol{\mathcal{O}}\}$. Since $\boldsymbol{P} = \boldsymbol{P} + \boldsymbol{\mathcal{O}}$, we have a map

$$\psi : \mathbb{F}_p \to C(\mathbb{F}_p) \setminus \{\boldsymbol{\mathcal{O}}\}, \bar{m} \mapsto \boldsymbol{P} = \left(\frac{\bar{m}^2 + \Delta}{\bar{m}^2 - \Delta}, \frac{2\bar{m}}{\bar{m}^2 - \Delta}\right) \tag{2}$$

where $\bar{m} = \Delta\, m$ and m is the slope of the line $\ell_{\boldsymbol{P},\boldsymbol{\mathcal{O}}}$ passing through $\boldsymbol{P}$ and $\boldsymbol{\mathcal{O}}$.[1] Note that $\bar{m}^2 - \Delta \neq 0$ for all $\bar{m} \in \mathbb{F}_p$ since Δ is a non-square in $\mathbb{F}_p$.

Proposition 1. *The set of solutions satisfying Eq.* (1) *is given by*

$$\big\{\psi(\bar{m}) \mid \bar{m} \in \mathbb{F}_p\big\} \cup \{\boldsymbol{\mathcal{O}}\} \ .$$

Proof. It is easy to see that ψ is injective. Indeed, assuming $\psi(\bar{m}_1) = \psi(\bar{m}_2)$, we get

$$\begin{aligned} & \begin{cases} (\bar{m}_1^2 + \Delta)(\bar{m}_2^2 - \Delta) = (\bar{m}_2^2 + \Delta)(\bar{m}_1^2 - \Delta) \\ 2\bar{m}_1(\bar{m}_2^2 - \Delta) = 2\bar{m}_2(\bar{m}_1^2 - \Delta) \end{cases} \\ \implies & \begin{cases} \bar{m}_1^2 = \bar{m}_2^2 \\ 2\bar{m}_1(\bar{m}_2^2 - \Delta) = 2\bar{m}_2(\bar{m}_1^2 - \Delta) \end{cases} \\ \implies & \ \bar{m}_1 = \bar{m}_2 \ . \end{aligned}$$

This concludes the proof by noting that there are $(p+1)$ solutions to Eq. (1). □

The inverse map is given by

$$\psi^{-1} : C(\mathbb{F}_p) \setminus \{\boldsymbol{\mathcal{O}}\} \to \mathbb{F}_p, (x, y) \mapsto \bar{m} = \frac{\Delta\, y}{x - 1} \ . \tag{3}$$

By augmenting $\mathbb{F}_p$ with ∞, maps ψ and ψ^{-1} yield an isomorphism $C(\mathbb{F}_p) \xrightarrow{\sim} \mathbb{F}_p \cup \{\infty\}$ by defining $\psi(\infty) = \boldsymbol{\mathcal{O}}$ and $\psi^{-1}(\boldsymbol{\mathcal{O}}) = \infty$.

We use this latter representation and define

$$\mathcal{T}_2(\mathbb{F}_p) = \big\{\bar{m} \mid \bar{m} = \psi^{-1}(x, y) \text{ with } (x, y) \in C(\mathbb{F}_p)\big\} \ . \tag{4}$$

[1] We consider $\bar{m}$ rather than m to get slightly faster arithmetic. This corresponds to the map presented in [21, § 5.2].

The neutral element in $\mathcal{T}_2(\mathbb{F}_p)$ is ∞. The inverse of $\bar{m}$ is $-\bar{m}$. Let $\bar{m}_1, \bar{m}_2 \in \mathcal{T}_2(\mathbb{F}_p) \setminus \{\infty\}$ and write $\otimes$ for the group law in $\mathcal{T}_2(\mathbb{F}_p)$. If $\bar{m}_1 = -\bar{m}_2$ then $\bar{m}_1 \otimes \bar{m}_2 = \infty$. If $\bar{m}_1 \neq -\bar{m}_2$, we get

$$\bar{m}_1 \otimes \bar{m}_2 = \psi^{-1}\big(\psi(m_1) \oplus \psi(m_2)\big) = \frac{\bar{m}_1\bar{m}_2 + \Delta}{\bar{m}_1 + \bar{m}_2} \ . \tag{5}$$

As a result, we can do cryptography in $\mathcal{T}_2(\mathbb{F}_p)$ by doing all arithmetic directly in $\mathbb{F}_p$.

2.2 Trace-based compression

The trace map is defined by

$$\mathrm{Tr} : \mathbb{F}_{p^2} \to \mathbb{F}_p : \alpha \mapsto \mathrm{Tr}(\alpha) = \alpha + \alpha^p \ .$$

Then $\alpha \in \mathsf{G}_{p,2}$ and its conjugate α^p are the roots of polynomial $(X - \alpha)(X - \alpha^p) = X^2 - \mathrm{Tr}(\alpha)X + 1$. Define $V_k = \mathrm{Tr}(\alpha^k)$. Since $V_k = \alpha^k + \alpha^{-k}$, it is easily verified that $V_{i+j} = V_iV_j - V_{i-j}$. In particular, we get $V_{2i} = {V_i}^2 - 2$ and $V_{2i+1} = V_{i+1}V_i - \mathrm{Tr}(\alpha)$. Therefore, if ℓ is the binary length of k, $\mathrm{Tr}(\alpha^k)$ can be quickly evaluated with only ℓ multiplications and ℓ squarings in $\mathbb{F}_p$ using the Montgomery ladder (e.g., see [13, Fig. 4]). As we see, trace-based representations are well suited for exponentiation.

Note that letting $P = \mathrm{Tr}(\alpha)$, $V_k = V_k(P, 1)$ corresponds to the k^{th} item of Lucas sequence $\{V_k(P, Q)\}$ with parameter $Q = 1$. Moreover, since $\alpha \in \mathsf{G}_{p,2}$, it follows that $\alpha \neq \alpha^p$ and $\Delta := \mathrm{Tr}(\alpha)^2 - 4 = P^2 - 4$ is a non-square. Let $\{U_k(P, 1)\}$ denote the companion Lucas sequence where $U_k \in \mathbb{F}_p$ satisfies $V_k + U_k\sqrt{\Delta} = 2\alpha^k$. Noting that $\sqrt{\Delta} = \alpha - \alpha^{-1}$, we have $U_k = (\alpha^k - \alpha^{-k})/(\alpha - \alpha^{-1})$. We also have ${V_k}^2 - \Delta\,{U_k}^2 = (V_k + U_k\sqrt{\Delta})(V_k - U_k\sqrt{\Delta}) = (2\alpha^k)(2\alpha^{-k}) = 4$. Consequently, an element $\alpha = x + y\sqrt{\Delta} \in \mathsf{G}_{p,2}$ can be equivalently written as $\alpha = \frac{V_1}{2} + \frac{U_1}{2}\sqrt{\Delta}$ and $\alpha^k = \frac{V_k}{2} + \frac{U_k}{2}\sqrt{\Delta}$ such that $\left(\frac{V_k}{2}\right)^2 - \Delta\left(\frac{U_k}{2}\right)^2 = 1$. In other words, we have $C(\mathbb{F}_p) = \left\{\left(\frac{V_k}{2}, \frac{U_k}{2}\right) \mid 0 \leq k \leq p\right\}$.

Enhanced trace-based representation. Trace-based representations over $\mathbb{F}_p$ can be 'enhanced' to allow the multiplication of two compressed elements. For $y \in \mathbb{F}_p$, we define the parity bit of y as $\mathrm{par}(y) = y \bmod 2$. As prime p is odd, we obviously have $\mathrm{par}(-y) = 1 - \mathrm{par}(y)$ if $y \in \mathbb{F}_p \setminus \{0\}$. Hence, a point $\boldsymbol{P} = (x, y) \in C(\mathbb{F}_p)$ is uniquely identified by the pair (x, β) where $\beta = \mathrm{par}(y)$. We call this the enhanced trace-based representation. Hence, being given (x_1, β_1) and (x_2, β_2) (corresponding to $\boldsymbol{P}_1$ and $\boldsymbol{P}_2 \in C(\mathbb{F}_p)$), the compressed value (x_3, β_3) (corresponding to $\boldsymbol{P}_3 = \boldsymbol{P}_1 \oplus \boldsymbol{P}_2$) can be obtained as follows: evaluate square roots $\sqrt{({x_1}^2 - 1)/\Delta}$ and $\sqrt{({x_2}^2 - 1)/\Delta}$ over $\mathbb{F}_p$; recover $\boldsymbol{P}_1 = (x_1, y_1)$ and $\boldsymbol{P}_2 = (x_2, y_2)$ from β_1 and β_2; compute $(x_3, y_3) = (x_1, y_1) \oplus (x_2, y_2)$; and output (x_3, β_3) with $\beta_3 = y_3 \bmod 2$. Compared to torus-based representation, this is however at the expense of the computation of two square roots and of further memory requirements.

2.3 Parametrization of higher dimensional tori

The next cases for which the ratio $r/\varphi(r)$ is large (and thus leading to optimal compression factors) are $r = 6$ and $r = 30$. An explicit compact representation of $\mathcal{T}_6(\mathbb{F}_p)$ is detailed in [21, Section 5.1]. For the case $r = 30$, we refer the reader to [8, Section 5].

3 Compact Representations over the Ring $\mathbb{Z}_N$

Let $N = pq$ be the product of two large primes. We let $\mathbb{Z}_N$ denote the ring of integers modulo N. The isomorphism $\mathbb{Z}_N \cong \mathbb{F}_p \times \mathbb{F}_q$ induces an isomorphism between $\mathcal{T}_r(\mathbb{Z}_N)$ and $\mathcal{T}_r(\mathbb{F}_p) \times \mathcal{T}_r(\mathbb{F}_q)$.

Current knowledge in cryptanalytic techniques implies that the hardness of factoring an RSA modulus N or computing discrete logarithms in a finite field of the size of N is broadly the same. Assuming that p and q are of equal size, the discrete logarithm problem in $\mathcal{T}_r(\mathbb{F}_p)$ and in $\mathcal{T}_r(\mathbb{F}_q)$ will thus not be easier than factoring N provided that $r \geq 2$ [12]. For efficiency reasons, a smaller value for r yields better performance. Henceforth, we focus on $\mathcal{T}_r(\mathbb{Z}_N)$ with $r = 2$.

3.1 Tori $\mathcal{T}_2(\mathbb{Z}_N)$ and $\widetilde{\mathcal{T}_2}(\mathbb{Z}_N)$

Consider the Pell equation over $\mathbb{Z}_N$,

$$C_{/\mathbb{Z}_N} : x^2 - \Delta\, y^2 = 1\,,$$

where $\Delta \in \mathbb{Z}_N^\times$ is a non-square modulo p and modulo q. By Chinese remaindering, the set of points $(x, y) \in \mathbb{Z}_N \times \mathbb{Z}_N$ satisfying this equation form a group, $C(\mathbb{Z}_N) = C(\mathbb{F}_p) \times C(\mathbb{F}_q)$, under the 'chord-and-tangent' law (see § 2.1). The neutral element is $\boldsymbol{\mathcal{O}} = (1, 0)$. For each point $\boldsymbol{P} \in C(\mathbb{Z}_N)$, there exists a unique pair of points $\boldsymbol{P}_p \in C(\mathbb{F}_p)$ and $\boldsymbol{P}_q \in C(\mathbb{F}_q)$ such that $\boldsymbol{P} \bmod p = \boldsymbol{P}_p$ and $\boldsymbol{P} \bmod q = \boldsymbol{P}_q$. We denote this equivalence by $\boldsymbol{P} = [\boldsymbol{P}_p, \boldsymbol{P}_q]$.

We can now extend the previous compression map (cf. Eq. (3)) to $\mathbb{Z}_N$. The only complication is that they are some points of the form $[\boldsymbol{P}_p, \boldsymbol{\mathcal{O}}_q]$ or $[\boldsymbol{\mathcal{O}}_p, \boldsymbol{P}_q]$. To deal more easily with these points, we consider a projective representation for the compressed result. We write $\bar{m}$ as a pair $(M : Z)$ and say that $\bar{m} = (M : Z)$ is equivalent to $\bar{m}' = (M' : Z')$ if there exists some $t \in \mathbb{Z}_N^\times$ such that $M' = t\,M$ and $Z' = t\,Z$. So we define

$$\psi^{-1} : C(\mathbb{Z}_N) \to \mathbb{P}^1(\mathbb{Z}_N), (x, y) \mapsto \bar{m} = (\Delta\, y : x - 1)\ . \tag{6}$$

This in turn leads to the definition of $\mathcal{T}_2(\mathbb{Z}_N)$,

$$\mathcal{T}_2(\mathbb{Z}_N) = \left\{\bar{m} \mid \bar{m} = \psi^{-1}(x, y) \text{ with } (x, y) \in C(\mathbb{Z}_N)\right\}\ . \tag{7}$$

Group law. We note $\otimes$ the group law on $\mathcal{T}_2(\mathbb{Z}_N)$. The neutral element is $(t:0)$ for some $t \in \mathbb{Z}_N^\times$. The inverse of an element $\bar{m} = (M:Z)$ is $(-M:Z)$. From Eq. (5), given $\bar{m}_1 = (M_1:Z_1)$ and $\bar{m}_2 = (M_2:Z_2)$ in $\mathcal{T}_2(\mathbb{Z}_N)$, a simple calculation shows that

$$(M_1:Z_1) \otimes (M_2:Z_2) = (M_1M_2 + \Delta\, Z_1Z_2 : M_1Z_2 + M_2Z_1) \ . \tag{8}$$

Observe that the group law is complete: it works for all inputs $\bar{m}_1, \bar{m}_2 \in \mathcal{T}_2(\mathbb{Z}_N)$.

Affine parametrization. The map given by Eq. (6) does not yield a compact representation for $\mathcal{T}_2(\mathbb{Z}_N)$ since each element $\bar{m}$ then requires two elements of $\mathbb{Z}_N$. A possible workaround is to ignore input points of the form $[\boldsymbol{P}_p, \boldsymbol{\mathcal{O}}_q]$ or $[\boldsymbol{\mathcal{O}}_p, \boldsymbol{P}_q]$ and to restrict to subset $\widetilde{C}(\mathbb{Z}_N) = \{(x,y) \in C(\mathbb{Z}_N) \mid x-1 \in \mathbb{Z}_N^\times\} \cup \{\boldsymbol{\mathcal{O}}\}$. A point $\boldsymbol{P} = (x,y) \in \widetilde{C}(\mathbb{Z}_N)$ corresponds to $\bar{m} = \left(\frac{\Delta\, y}{x-1} : 1\right)$ if $\boldsymbol{P} \neq \boldsymbol{\mathcal{O}}$, and $\bar{m} = (1:0)$ otherwise. We define

$$\widetilde{\mathcal{T}}_2(\mathbb{Z}_N) = \left\{\bar{m} \mid \bar{m} = \psi^{-1}(x,y) \text{ with } (x,y) \in \widetilde{C}(\mathbb{Z}_N)\right\} \ . \tag{9}$$

From the above observation, an element $\bar{m}$ in $\widetilde{\mathcal{T}}_2(\mathbb{Z}_N)$ can be represented by an element of $\mathbb{Z}_N$ plus one bit: $\left(\frac{\Delta\, y}{x-1}, 1\right)$ or $(1,0)$. Yet another possibility is to represent $\bar{m}$ as an element of $\mathbb{A}^1(\mathbb{Z}_N) \cup \{\infty\}$. Namely, if $\boldsymbol{P} = (x,y) \in \widetilde{C}(\mathbb{Z}_N)$ then $\bar{m} = \frac{\Delta\, y}{x-1}$ if $\boldsymbol{P} \neq \boldsymbol{\mathcal{O}}$, and $\bar{m} = \infty$ otherwise. In both cases, we get a compact representation for $\widetilde{\mathcal{T}}_2(\mathbb{Z}_N)$.

The group $\mathcal{T}_2(\mathbb{Z}_N)$ consists of all the elements of $\widetilde{\mathcal{T}}_2(\mathbb{Z}_N)$ together with a number of elements of the form $(M:Z)$ with $\gcd(Z,N) = p$ or q (corresponding to points $[\boldsymbol{P}_p, \boldsymbol{\mathcal{O}}_q]$ and $[\boldsymbol{\mathcal{O}}_p, \boldsymbol{P}_q]$ in $C(\mathbb{Z}_N)$). The 'chord-and-tangent' law on $\widetilde{C}(\mathbb{Z}_N)$, whenever it is defined, coincides with the group law on $C(\mathbb{Z}_N) = C(\mathbb{F}_p) \times C(\mathbb{F}_q)$. The same holds for $\widetilde{\mathcal{T}}_2(\mathbb{Z}_N)$. In practice, for cryptographic applications, N is the product of two large primes. It is therefore extremely unlikely that operation $\otimes$ is not defined on $\widetilde{\mathcal{T}}_2(\mathbb{Z}_N)$.

3.2 Torus-based vs. trace-based compression

Similarly to § 2.2, Lucas sequences can be defined over the ring $\mathbb{Z}_N$ by Chinese remaindering. Trace-based or equivalently Lucas-based compressions are well suited to exponentiation. For example, Smith and Lennon proposed in [27] an analogue to RSA using Lucas sequence $\{V_k(P,1)\}$ over $\mathbb{Z}_N$.

When more than a mere exponentiation is required, trace-based representations are not applicable. Indeed, let $\boldsymbol{P}_1 = (x_1, y_1), \boldsymbol{P}_2 = (x_2, y_2) \in C(\mathbb{Z}_N)$. Computing $\boldsymbol{P}_3 = \boldsymbol{P}_1 \oplus \boldsymbol{P}_2$ being given $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ is easy: we have $\boldsymbol{P}_3 = (x_1x_2 + \Delta\, y_1y_2, x_1y_2 + x_2y_1)$. However, computing $x_3 = x_1x_2 + \Delta\, y_1y_2$ being only given x_1 and x_2 is not possible.

Even an enhanced trace-based representation (cf. § 2.2) does not seem helpful when working over $\mathbb{Z}_N$. Here is an example of such an enhanced compression for

Blum integers N (i.e., $N = pq$ with primes $p, q \equiv 3 \pmod 4$). As before, for $y \in \mathbb{Z}_N$, we define $\mathrm{par}(y) = y \bmod 2$. We also define $\mathrm{chr}(y) = 0$ if $\left(\frac{y}{N}\right) = 1$ and $\mathrm{chr}(y) = 1$ otherwise, where $\left(\frac{y}{N}\right)$ denotes the Jacobi symbol of y modulo N. Since $p, q \equiv 3 \pmod 4$, we have $\left(\frac{-1}{p}\right) = \left(\frac{-1}{q}\right) = -1$. It is therefore easily verified that a point $\boldsymbol{P} = (x, y) \in C(\mathbb{Z}_N)$ is uniquely identified by the tuple (x, β, χ) where $\beta = \mathrm{par}(y)$ and $\chi = \mathrm{chr}(y)$, that is, with one element of $\mathbb{Z}_N$ and two bits. Unfortunately, decompressing (x, β, χ) into $\boldsymbol{P} = (x, y)$ requires the knowledge of p and q, which are, in most settings, private values. Unlike the finite field case, we do not know enhanced trace-based representation over $\mathbb{Z}_N$ allowing to multiply compressed elements. Only torus-based representation over $\mathbb{Z}_N$ is available in this case to get a compact representation.

3.3 Extensions and generalizations

Because the problems of computing discrete logarithms and of factoring were assumed to be balanced for an RSA modulus $N = pq$, we focused on the case $\mathcal{T}_2(\mathbb{Z}_N)$. But the same methodology extends to higher-dimensional tori. It also generalizes to more general moduli; for example, to RSA moduli made of three prime factors. This allows for different trade-offs between the two computational problems.

4 Applications

Our compression technique reduces the parameter size (typically by a factor of two). This in turn reduces the requirements for storage and transmission. It saves a significant amount in applications where many group elements are evaluated.

As an example, we consider the ACJT group signature scheme. We pointed out in the introduction (see also [4]) that the group manager in the original scheme may know the factors of RSA modulus $N = pq$ and so can frame group members if the computation of discrete logarithms modulo the factors of N is feasible. As will be apparent, a torus-based implementation allows one to keep the security of the original ACJT scheme even when the group manager is not entirely trustworthy — without increasing the length of RSA modulus N.

To simplify the presentation, we omit the various security lengths ($\lambda_1, \lambda_2, \gamma_1, \gamma_2$) and corresponding ranges (Λ, Γ). We refer the reader to [1] for details. Slight modifications need to be brought. The original ACJT group signature scheme makes use of a strong RSA modulus, that is, $N = pq$ with $p = 2p' + 1$ and $q = 2q' + 1$ for primes p', q'. Since $\mathsf{G}_{p,2}$ (resp. $\mathsf{G}_{q,2}$) has order $p + 1$ (resp. $q + 1$), we choose an RSA modulus $N = pq$ with $p = 4p' - 1$ and $q = 4q' - 1$ for primes p', q'; note that doing so -1 is a non-square modulo p and modulo q (i.e., $p, q \equiv 3 \pmod 4$), which yields faster arithmetic. We let $\mathbb{T}_2$ denote the subgroup of order $p'q'$ in $\mathcal{T}_2(\mathbb{Z}_N)$. Finally, we let $\widetilde{\mathbb{T}}_2 = \mathbb{T}_2 \cap \widetilde{\mathcal{T}}_2(\mathbb{Z}_N)$.

Being a group signature scheme, our modified scheme consists of five algorithms. We use the notation of [1].

Setup Select two random primes p', q' such that $p = 4p' - 1$ and $q = 4q' - 1$ are prime. Set the modulus $N = pq$. Choose random elements a, a_0, g, h in $\widetilde{\mathbb{T}}_2$. Choose a random element $x \in \mathbb{Z}^{\times}_{p'q'}$ and set $y = g^x \in \widetilde{\mathbb{T}}_2$.
The group public key is $\mathcal{Y} = (N, a, a_0, y, g, h)$. The corresponding secret key (known only to the group manager) is $\mathcal{S} = (p', q', x)$.

Join Each user U_i interactively constructs with the group manager a membership certificate $[A_i, e_i]$ satisfying ${A_i}^{e_i} = a^{x_i} \otimes a_0$ in $\widetilde{\mathbb{T}}_2$ for some prime e_i. Parameter x_i is the private key of U_i (and is unknown to the group manager).

Sign Generate a random value w and compute in $\widetilde{\mathbb{T}}_2$

$$T_1 = A_i \otimes y^w, \;\; T_2 = g^w, \;\; T_3 = g^{e_i} \otimes h^w \;\; .$$

Randomly choose values r_1, r_2, r_3, r_4 and compute

1. $d_1 = {T_1}^{r_1} \otimes (a^{r_2} \otimes y^{r_3})^{-1}$, $d_2 = {T_2}^{r_1} \otimes (g^{r_3})^{-1}$, $d_3 = g^{r_4}$ and $d_4 = g^{r_1} \otimes h^{r_4}$ (all in $\widetilde{\mathbb{T}}_2$);
2. $c = \mathcal{H}(\mathcal{Y} \| T_1 \| T_2 \| T_3 \| d_1 \| d_2 \| d_3 \| d_4 \| m)$ where m is the message being signed;
3. $s_1 = r_1 - c(e_i - 2^{\gamma_1})$, $s_2 = r_2 - c(x_i - 2^{\lambda_1})$, $s_3 = r_3 - ce_i w$ and $s_4 = r_4 - cw$ (all in $\mathbb{Z}$).

The signature on message m is $\sigma = (c, s_1, s_2, s_3, s_4, T_1, T_2, T_3)$.

Verify Compute in $\widetilde{\mathbb{T}}_2$

$$d'_1 = {a_0}^c \otimes {T_1}^{s_1 - c2^{\gamma_1}} \otimes (a^{s_2 - c2^{\lambda_1}} \otimes y^{s_3})^{-1}, \;\; d'_2 = {T_2}^{s_1 - c2^{\gamma_1}} \otimes (g^{s_3})^{-1},$$
$$d'_3 = {T_2}^c \otimes g^{s_4}, \;\; d'_4 = {T_3}^c \otimes g^{s_1 - c2^{\gamma_1}} \otimes h^{s_4} \;\; .$$

Accept the signature if and only if $c' := \mathcal{H}(\mathcal{Y} \| T_1 \| T_2 \| T_3 \| d'_1 \| d'_2 \| d'_3 \| d'_4 \| m)$ is equal to c (and if the signature components belong to appropriate ranges).

Open Check the signature's validity. The group manager then recovers $A_i = T_1 \otimes ({T_2}^x)^{-1}$ (in $\widetilde{\mathbb{T}}_2$).

We now discuss the performance of our modified scheme and compare it with the original ACJT scheme.

Let ℓ_N denote the binary length of modulus N. The system secret key $\mathcal{S}$ requires $2\ell_N$ bits. As shown in § 3.1, an element in $\widetilde{\mathbb{T}}_2 \setminus \{\infty\}$ can be coded with ℓ_N bits using an affine parametrization. Hence, the common public key $\mathcal{Y}$ consisting of 6 elements of $\widetilde{\mathbb{T}}_2$ requires $6\ell_N$ bits. The size of exponent e_i in membership certificate $[A_i, e_i]$ and of corresponding private key x_i are about the size of N^2; therefore, a membership certificate requires roughly $3\ell_N$ bits and the user's private key roughly $2\ell_N$ bits. Since the size of s_j $(1 \leq j \leq 4)$ is about the

size of N^2, a signature $\sigma = (c, s_1, s_2, s_3, s_4, T_1, T_2, T_3)$ requires approximatively $11\ell_N$ bits. Typically, for a 80-bit security level (i.e., 2048-bit modulus for the ACJT scheme and 1024-bit modulus for its torus-based implementation), we have:

Table 1. Performance comparison: Typical lengths.

	ACJT scheme	Torus-based scheme
Common public key	12 kb	6 kb
System secret key	4 kb	2 kb
Membership certificate	6 kb	3 kb
User's private key	4 kb	2 kb
Signature (approx.)	22 kb	11 kb

Our torus-based signatures are not only shorter, as we will see, they are also faster to generate. We neglect the cost of additions and hash computations. For the sake of comparison, we assume that exponentiations are done with the basic square-and-multiply algorithm and that multi-exponentiations are evaluated with the simultaneous binary exponentiation algorithm (see e.g. [19, Algorithm 14.88]). A k-exponentiation with exponent of binary length ℓ then amounts to $(\ell - 1) \cdot (\mathsf{S} + \frac{2^k-1}{2^k}\mathsf{M})$, on average, where S and M represent the cost of a squaring and of a multiplication in $\widetilde{\mathbb{T}}_2$. It also requires $(2^k - 2)\mathsf{M}$ for the precomputation. Since T_1, T_2 involve exponents of size about ℓ_N bits and T_3, d_1, d_2, d_3, d_4 involve exponents of size about $2\ell_N$ bits, the generation of a signature takes about

$$\begin{aligned}&\ell_N(\mathsf{S} + \tfrac{1}{2}\mathsf{M}) + \ell_N(\mathsf{S} + \tfrac{1}{2}\mathsf{M}) + 2\ell_N(\mathsf{S} + \tfrac{3}{4}\mathsf{M}) + 2\ell_N(\mathsf{S} + \tfrac{7}{8}\mathsf{M})\\ &\quad + 2\ell_N(\mathsf{S} + \tfrac{3}{4}\mathsf{M}) + 2\ell_N(\mathsf{S} + \tfrac{1}{2}\mathsf{M}) + 2\ell_N(\mathsf{S} + \tfrac{3}{4}\mathsf{M}) = 12\ell_N\mathsf{S} + 8.25\ell_N\mathsf{M}\,, \quad (10)\end{aligned}$$

neglecting the precomputation.

In our scheme, we have $p, q \equiv 3 \pmod 4$. We can thus take $\Delta = -1$. In this case, using projective coordinates, the multiplication of two elements $\bar{m}_1 = (M_1 : Z_1)$ and $\bar{m}_2 = (M_2 : Z_2)$ in $\widetilde{\mathbb{T}}_2$, $\bar{m}_3 = \bar{m}_1 \otimes \bar{m}_2$, simplifies to $\bar{m}_3 = (M_3 : Z_3)$ with $M_3 = M_1M_2 + Z_1Z_2$ and $Z_3 = M_1Z_2 + M_2Z_1 = (M_1 + Z_1)(M_2 + Z_2) - M_3$. Let s and m denote the cost of a square and a multiplication in $\mathbb{Z}_N$. The multiplication of two elements of $\widetilde{\mathbb{T}}_2$ requires thus $3\mathsf{m}$. Note that for a mixed multiplication (i.e., when one of the two operands has its Z-coordinate equal to 1), the cost reduces to $2\mathsf{m}$. Squaring $\bar{m}_1 = (M_1 : Z_1)$ can be evaluated as $\bar{m}_3 = (M_3 : Z_3)$ with $Z_3 = 2M_1Z_1$ and $M_3 = (M_1 + Z_1)^2 - Z_3$ and requires thus $1\mathsf{s} + 1\mathsf{m}$. If the precomputed values in the k-exponentiation are expressed in affine way (this can be done with a single inversion and a few multiplications in $\mathbb{Z}_N$ using the so-called Montgomery's trick), we have $\mathsf{M} = 2\mathsf{m}$ and $\mathsf{S} = 1\mathsf{s} + 1\mathsf{m}$. Therefore, neglecting the cost of this inversion in $\mathbb{Z}_N$ and assuming $\mathsf{s} = 0.8\mathsf{m}$, we obtain that the cost of a torus-based ACJT group signature is about $(12 \cdot 1.8 + 8.25 \cdot 2)\ell_N\mathsf{m} = 38.1\,\ell_N\mathsf{m}$.

Similarly, from Eq. (10), we obtain that the cost of a regular ACJT group signature is about $(12 \cdot 0.8 + 8.25)\ell_N \mathsf{m} = 17.85\,\ell_N \mathsf{m}$, assuming again $\mathsf{s} = 0.8\mathsf{m}$. But since the length of ℓ_N is twice smaller in $\widetilde{T}_2$, the expected speed-up factor amounts to

$$\frac{17.85\,\ell_N \cdot (\ell_N)^2}{38.1\,(\ell_N/2) \cdot (\ell_N/2)^2} \approx 3.75 \ .$$

In practice, the expected speed-up factor is even more spectacular as the above value assumes that the same exponentiation algorithms are being used; however, for the same amount of memory, the torus-based implementation can be sped up using more pre-computed values and higher-order methods. Note also that the above analysis neglects the cost of inversion in $\mathbb{Z}_N^\times$ (in the evaluation of d_1 and d_2) for the regular ACJT signatures.

5 Conclusion

This paper extended the concept of torus-based cryptography over the ring of integers modulo N. Our extended setting finds applications in cryptographic schemes whose security is related to factoring and discrete logarithms. Typically, it results in twice shorter keys and outputs and offers faster computation. This was exemplified with a torus-based implementation of the ACJT group signature scheme.

References

1. Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In M. Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 255–270. Springer-Verlag, 2000.
2. Niko Barić and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In W. Fumy, editor, *Advances in Cryptology – EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 480–494. Springer-Verlag, 1997.
3. Dan Boneh. The decision Diffie-Hellman problem. In J. Buhler, editor, *Algorithmic Number Theory (ANTS-III)*, volume 1423 of *Lecture Notes in Computer Science*, pages 48–63. Springer-Verlag, 1998.
4. Jan Camenisch and Jens Groth. Group signatures: Better efficiency and new theoretical aspects. In C. Blundo and S. Cimato, editors, *Security in Communication Networks (SCN 2004)*, volume 3352 of *Lecture Notes in Computer Science*, pages 120–133. Springer-Verlag, 2004.
5. David Chaum and Eugène van Heyst. Group signatures. In D.W. Davies, editor, *Advances in Cryptology – EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer-Verlag, 1991.
6. Isabelle Déchène. *Generalized Jacobians in Cryptography.* PhD thesis, McGill University, Montreal, Canada, 2005.
7. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

8. Marten van Dijk, Robert Granger, Dan Page, Karl Rubin, Alice Silverberg, Martijn Stam, and David Woodruff. Practical cryptography in high dimensional tori. In R. Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3492 of *Lecture Notes in Computer Science*, pages 234–250. Springer-Verlag, 2005.
9. Marten van Dijk and David Woodruff. Asymptotically optimal communication for torus-based cryptography. In M.K. Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 157–178. Springer-Verlag, 2004.
10. Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero-knowledge protocols to prove modular polynomial equations. In B. Kaliski, Jr, editor, *Advances in Cryptology – CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 16–30. Springer-Verlag, 1997.
11. Robert Granger, Dan Page, and Martijn Stam. A comparison of CEILIDH and XTR. In D.A. Buell, editor, *Algorithmic Number Theory (ANTS-VI)*, volume 3076 of *Lecture Notes in Computer Science*, pages 235–249. Springer-Verlag, 2004.
12. Robert Granger and Frederik Vercauteren. On the discrete logarithm problem on algebraic tori. In V. Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 66–85. Springer-Verlag, 2005.
13. Marc Joye and Sung-Ming Yen. The Montgomery powering ladder. In B.S. Kaliski, Jr., Ç.K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2000*, volume 2523 of *Lecture Notes in Computer Science*, pages 291–302. Springer-Verlag, 2000.
14. Franz Lemmermeyer. Higher descent on Pell conics (III). Preprint, 2003.
15. Arjen K. Lenstra. Using cyclotomic polynomials to construct efficient discrete logarithm cryptosystems over finite fields. In V. Varadharajan, J. Pieprzyk, and Y. Mu, editors, *Information Security and Privacy (ACISP '97)*, volume 1270 of *Lecture Notes in Computer Science*, pages 127–138. Springer-Verlag, 1997.
16. Arjen K. Lenstra and Eric R. Verheul. The XTR public key system. In M. Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, page 119. Springer-Verlag, 2000.
17. Kevin S. McCurley. A key distribution system equivalent to factoring. *Journal of Cryptology*, 1(2):95–105, 1988.
18. Alfred J. Menezes. *Elliptic Curve Public Key Cryptosystems*. Kluwer Academic Publishers, 1993.
19. Alfred J. Menezes, Paul C. van Oorchot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.
20. Boleslas Niewenglowski. Note sur les équations $x^2 - ay^2 = 1$ et $x^2 - ay^2 = -1$. *Bulletin de la Société Mathématique de France*, 35:126–131, 1907.
21. Karl Rubin and Alice Silverberg. Torus-based cryptography. In D. Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 349–365. Springer-Verlag, 2003.
22. Karl Rubin and Alice Silverberg. Using primitive subgroups to do more with fewer bits. In D.A. Buell, editor, *Algorithmic Number Theory (ANTS VI)*, volume 3076 of *Lecture Notes in Computer Science*, pages 18–41. Springer-Verlag, 2004.
23. Karl Rubin and Alice Silverberg. Compression in finite fields and torus-based cryptography. *SIAM Journal on Computing*, 37(5):1401–1428, 2008.
24. Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

25. Zahava Shmuely. Composite Diffie-Hellman public key generating systems hard to break. Technical Report 356, Israel Institute of Technology, Computer Science Department, Technion, February 1985.
26. Peter Smith and Christopher Skinner. A public-key cryptosystem and a digital signature system based on the Lucas function analogue to discrete logarithms. In J. Pieprzyk and R. Safavi-Naini, editors, *Advances in Cryptology – ASIACRYPT '94*, volume 917 of *Lecture Notes in Computer Science*, pages 357–364. Springer-Verlag, 1995.
27. Peter J. Smith and Michael J.J. Lennon. LUC: A new public key system. In E.G. Dougall, editor, *9th International Conference on Information Security (IFIP/Sec '93)*, volume A-37 of *IFIP Transactions*, pages 103–117. North-Holland, 1993.
28. Martijn Stam and Arjen K. Lenstra. Speeding up XTR. In C. Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 125–143. Springer-Verlag, 2001.
29. Trusted Computing Group. TCG TPM specification 1.2. Available at `http://www.trustedcomputinggroup.org/`, 2003.