The XTR public key system

(extended version of Crypto 2000 presentation)

Arjen K. Lenstra
Citibank, New York
Technical University Eindhoven

Eric R. Verheul
PricewaterhouseCoopers

XTR

stands for ECSTR

Efficient Compact Subgroup Trace Representation

Overview

- XTR background
- XTR security
- Comparison to traditional representation, RSA, and ECC
- XTR subgroup representation
- XTR subgroup exponentiation
- XTR multi−exponentiation
- XTR parameter generation
- Improved XTR parameter generation
- XTR application example
- Disadvantages?
- Related work
- Conclusion

XTR is **not** a new cryptosystem

- XTR is a traditional subgroup Discrete Logarithm system

- XTR uses an efficient and compact method
  to represent subgroup elements (like LUC, but better)

- XTR removes the distinction between conjugates (like LUC)

- The security of XTR is based on the Discrete Logarithm problem
  in the subgroup of $GF(p^6)^*$ of order dividing $p^2 - p + 1$
  (LUC uses the subgroup of $GF(p^2)^*$ of order dividing $p + 1$)

Subgroups of $GF(p^t)^*$

- $\# \, GF(p^t)^* = \prod_{d|t} \phi_d(p)$, $\phi_d(X)$ is the dth cyclotomic polynomial

  with Pohlig–Hellman:

  computing Discrete Logarithms in $GF(p^t)^*$ is equivalent to
  computing Discrete Logarithms in all order $\phi_d(p)$ subgroups
- for d dividing t with $d < t$:

  the order $\phi_d(p)$ subgroup can efficiently be embedded in the

  multiplicative group $GF(p^d)^*$ of true subfield $GF(p^d)$ of $GF(p^t)$

$\Rightarrow$ according to current (published) state of the art:

for d dividing t with $d < t$ the DL problem in

the order $\phi_d(p)$ subgroups is easier than DL problem in $GF(p^t)^*$

$\Rightarrow$ in general: the DL problem in the order $\phi_t(p)$ subgroup

is as hard as the DL problem in $GF(p^t)^*$

Subgroups of $GF(p^6)^*$

$p^6 - 1 = (p - 1)(p + 1)(p^2 + p + 1)(p^2 - p + 1)$

- Subgroup of order $p - 1$ can be embedded in $GF(p)^*$
- Subgroup of order $p + 1$ can be embedded in $GF(p^2)^*$
- Subgroup of order $p^2 + p + 1$ can be embedded in $GF(p^3)^*$
- Subgroup of order $\phi_6(p) = p^2 - p + 1$ cannot be embedded in $GF(p^t)^*$ for $t = 1, 2, 3$

$\Rightarrow$ (Pohlig–Hellman) order $p^2 - p + 1$ subgroup is as hard as $GF(p^6)^*$,

or: if order $p^2 - p + 1$ subgroup is easier than $GF(p^6)^*$

then $GF(p^6)^*$ is at most as hard as $GF(p^3)^*$ (and that is unlikely)

XTR security

- XTR versions of cryptographic protocols provably as secure as traditional versions over GF(p^6)
- either XTR is secure (because GF(p^6) is secure) or XTR is not secure (and thus GF(p^6) is not secure)
- current state of the art:

  Discrete Logarithms in GF(p^6)* are
  at least as hard as (or harder than) Discrete Logarithms in
  multiplicative group of $6\log_2(p)$–bit prime field

In general no additional risk in moving
from prime fields to extension fields of comparable size,
as long as subgroup order divides $\phi_t(p)$ (in GF(p^t)*, p large)

Comparison of traditional and XTR representation

$<g> \subset GF(p^6)^*$, g of prime order q dividing $p^2 - p + 1$

	Traditional	XTR
Bits to represent g^m	$6*\log_2(p)$	$2*\log_2(p)$
Multiplications in GF(p) to compute g^m	$21*\log_2(m)$	$8*\log_2(m)$

(order $\approx q$ subgroup of $\approx 6*\log_2(p)$–bit prime field are even slower)

Comparison of traditional and XTR representation

$\langle g \rangle \subset GF(p^6)^*$, g of prime order q dividing $p^2 - p + 1$, $h \in \langle g \rangle$

	Traditional	XTR
Bits to represent g^m, $g^m * h^n$	$6*\log_2(p)$	$2*\log_2(p)$
Multiplications in GF(p) to compute g^m,	$21*\log_2(m)$	$8*\log_2(m)$
$g^m * h^n$ with $m \approx n$	$25.5*\log_2(m)$	$16*\log_2(m)$

XTR, RSA comparison

Run times in milliseconds on 450MHz Pentium II NT, using generic sofware implementation

	170–bit XTR	1020–bit RSA
Parameter/Key selection	73 ms	1224 ms
Encrypting/Verifying	23 ms	5 ms for 32–bit *e*
Decrypting/Signing	11 ms	40 ms (no CRT: 123 ms)
Public Key size	680 bits	1050 bits
ID–based Public Key size	388 bits	510 bits

XTR, ECC comparison (for ECC over prime fields)

Run time estimates (based on multiplication count in GF(p);
from Cohen/Miyaji/Ono Asiacrypt'98 paper)

	170–bit XTR	170–bit ECC
Parameter/Key selection	73 ms	hours ?
Encrypting	23 ms (2720)	28 ms (3400)
Decrypting	11 ms (1360)	16 ms (1921)
Signing	11 ms (1360)	14 ms (1700)
Verifying	23 ms (2754)	$\leq$ 21 ms (2575)
Public Key size	680 bits	766 bits
ID–based Public Key size	388 bits	304 bits
Shared Public Key size	340 bits	171 bits

How does it work?

XTR subgroup element representation

$<g> \subset \mathrm{GF}(p^6)^*$, g of prime order q dividing $p^2 - p + 1$, $q > 3$

- Let $Tr(g) = g + g^{p^2} + g^{p^4} \in \mathrm{GF}(p^2)$ be the trace over $\mathrm{GF}(p^2)$ of g

- Let $F(c,X) = X^3 - cX^2 + c^p X - 1$, for $c \in \mathrm{GF}(p^2)$

- Then $F(Tr(g),g) = 0$

$\Rightarrow$ g and its conjugates can be represented by $Tr(g) \in \mathrm{GF}(p^2)$

XTR subgroup exponentiation

$<g> \subset \mathrm{GF}(p^6)^*$, g of prime order q dividing $p^2 - p + 1$, $q > 3$

$$F(Tr(g^n), g^n) = g^{3n} - Tr(g^n)\, g^{2n} + Tr(g^n)^p\, g^n - 1 = 0$$

$$\Rightarrow Tr(g^{m+n}) = Tr(g^n) * Tr(g^m) - Tr(g^n)^p * Tr(g^{m-n}) + Tr(g^{m-2n})$$

XTR subgroup exponentiation

$<g> \subset \mathrm{GF}(p^6)^*$, g of prime order q dividing $p^2 - p + 1$, $q > 3$

$$F(Tr(g^n), g^n) = g^{3n} - Tr(g^n)\, g^{2n} + Tr(g^n)^p\, g^n - 1 = 0$$

$$\Rightarrow \quad g^{3n} = Tr(g^n)\, g^{2n} - Tr(g^n)^p\, g^n + 1$$

multiply by g^{m-2n}

$$\Rightarrow \quad g^{m+n} = Tr(g^n)\, g^m - Tr(g^n)^p g^{m-n} + g^{m-2n}$$

add this to its p^2th and p^4th power

$$\Rightarrow Tr(g^{m+n}) = Tr(g^n) * Tr(g^m) - Tr(g^n)^p * Tr(g^{m-n}) + Tr(g^{m-2n})$$

XTR subgroup exponentiation

$<g> \subset \mathrm{GF}(p^6)^*$, g of prime order q dividing $p^2 - p + 1$, $q > 3$

$$F(Tr(g^n), g^n) = g^{3n} - Tr(g^n)\, g^{2n} + Tr(g^n)^p\, g^n - 1 = 0$$

$$\Rightarrow Tr(g^{m+n}) = Tr(g^n)*Tr(g^m) - Tr(g^n)^p*Tr(g^{m-n}) + Tr(g^{m-2n})$$

Thus:

$$Tr(g^{2n}) = Tr(g^n)^2 - 2Tr(g^n)^p$$

$$Tr(g^{n+2}) = Tr(g)*Tr(g^{n+1}) - Tr(g)^p*Tr(g^n) + Tr(g^{n-1})$$

$$Tr(g^{2n-1}) = Tr(g^n)*Tr(g^{n-1}) - Tr(g^n)^p*Tr(g)^p + Tr(g^{n+1})^p$$

$$Tr(g^{2n+1}) = Tr(g^n)*Tr(g^{n+1}) - Tr(g^n)^p*Tr(g) + Tr(g^{n-1})^p$$

XTR subgroup exponentiation, continued

- $p \equiv 2 \bmod 3$, α with $\alpha^2 + \alpha + 1 = (\alpha^3 - 1)/(\alpha - 1) = 0$, then

  $\{\alpha, \alpha^p\} = \{\alpha, \alpha^2\}$ forms normal basis for GF(p^2) over GF(p)
- $(x_1\alpha + x_2\alpha^2)^p = x_2\alpha + x_1\alpha^2$: pth powering in GF(p^2) is free

Thus, given $Tr(g)$ and $Tr(g^n)$,

$$Tr(g^{2n}) = Tr(g^n)^2 - 2Tr(g^n)^p$$

takes two GF(p) multiplications and, with $Tr(g^{n+1})$, $Tr(g^{n-1})$,

$$Tr(g^{n+2}) = Tr(g)*Tr(g^{n+1}) - Tr(g)^p*Tr(g^n) + Tr(g^{n-1})$$

$$Tr(g^{2n-1}) = Tr(g^n)*Tr(g^{n-1}) - Tr(g^n)^p*Tr(g)^p + Tr(g^{n+1})^p$$

$$Tr(g^{2n+1}) = Tr(g^n)*Tr(g^{n+1}) - Tr(g^n)^p*Tr(g) + Tr(g^{n-1})^p$$

take four GF(p) multiplications each

XTR subgroup exponentiation, continued

Given $Tr(g)$ and $(Tr(g^{2n}), Tr(g^{2n+1}), Tr(g^{2n+2}))$

it takes eight multiplications in GF(p) to compute

$(Tr(g^{4n}), Tr(g^{4n+1}), Tr(g^{4n+2}))$ ‘bit off’

or (of $(m-1)/2$)

$(Tr(g^{4n+2}), Tr(g^{4n+3}), Tr(g^{4n+4}))$ ‘bit on’

$\Rightarrow$ computing $Tr(g^m)$ given $Tr(g)$
takes $8\log_2(m)$ multiplications in GF(p)

iteration different from ordinary ‘multiply and square’:
‘bit off’ and ‘bit on’ computations are almost the same

XTR multi–exponentiation (signature verification)

Given $Tr(g)$ and $Tr(g^k)$ for a secret k, compute $Tr(g^m * g^{kn})$

- compute $e = m/n$ modulo q
- compute $(Tr(g^{e-1}), Tr(g^e), Tr(g^{e+1}))$
- compute $V = \begin{pmatrix} Tr(g^{-2}) & Tr(g^{-1}) & 3 \\ Tr(g^{-1}) & 3 & Tr(g) \\ 3 & Tr(g) & Tr(g^2) \end{pmatrix}^{-1} \begin{pmatrix} Tr(g^{e-1}) \\ Tr(g^e) \\ Tr(g^{e+1}) \end{pmatrix}$

$$V = \frac{1}{D}\begin{pmatrix} 2c^2-6c^p & 2c^{2p}+3c-c^{p+2} & c^{p+1}-9 \\ 2c^{2p}+3c-c^{p+2} & (c^2-2c^p)^{p+1}-9 & (2c^{2p}+3c-c^{p+2})^p \\ c^{p+1}-9 & (2c^{2p}+3c-c^{p+2})^p & (2c^2-6c^p)^p \end{pmatrix}\begin{pmatrix} Tr(g^{e-1}) \\ Tr(g^e) \\ Tr(g^{e+1}) \end{pmatrix}$$

with $D = c^{2p+2} + 18c^{p+1} - 4(c^{3p} + c^3) - 27 \in \mathrm{GF}(p)$ and $c = Tr(g)$

XTR multi–exponentiation (signature verification)

Given $Tr(g)$ and $Tr(g^k)$ for a secret k, compute $Tr(g^m * g^{kn})$

- compute $e = m/n$ modulo q
- compute $(Tr(g^{e-1}), Tr(g^e), Tr(g^{e+1}))$
- compute $V = \begin{pmatrix} Tr(g^{-2}) & Tr(g^{-1}) & 3 \\ Tr(g^{-1}) & 3 & Tr(g) \\ 3 & Tr(g) & Tr(g^2) \end{pmatrix}^{-1} \begin{pmatrix} Tr(g^{e-1}) \\ Tr(g^e) \\ Tr(g^{e+1}) \end{pmatrix}$
- compute $Tr(g^{e+k}) = (Tr(g^{k-1}), Tr(g^k), Tr(g^{k+1})) * V$

need 'neighbors' of $Tr(g^k)$ too,

- compute $Tr(g^{(e+k)n}) = Tr(g^m * g^{kn})$ else k is not well–defined

XTR parameter generation

find primes $p \equiv 2 \bmod 3$ and $q > 3$ with q dividing $p^2 - p + 1$,
and $Tr(g)$ for g of order q (no need to compute g itself)

- find r such that $r^2 - r + 1$ is prime, let $q = r^2 - r + 1$,
  find k such that $r + k*q$ is prime (and 2 mod 3), let $p = r + k*q$
- pick a $c \in \mathrm{GF}(p^2)$,
  assume: $c = Tr(h)$ for h of order dividing $p^2 - p + 1$,
  compute $Tr(h^{p+1})$ using XTR exponentiation, then:
  assumption correct $\Leftrightarrow$ $Tr(h^{p+1}) \in \mathrm{GF}(p^2)\backslash\mathrm{GF}(p)$,
- on average 3 trials for c suffice

- compute $Tr(g) = Tr(h^{(p^2-p+1)/q})$; pick new c if $Tr(g) = 3$

$\Rightarrow$ XTR parameter generation takes on average $(3*8+8)\log_2(m)$ multiplications in $\mathrm{GF}(p)$ (plus the time to generate q and p)
and: no additional software on top of XTR arithmetic

Improved XTR parameter generation

Finding c such that $c = Tr(h)$ for h of order dividing $p^2 - p + 1$

$\Leftrightarrow$ $F(c,X)$ irreducible over $\mathrm{GF}(p^2)[X]$ $\Leftrightarrow$

- $Tr(h^{p+1}) \in \mathrm{GF}(p^2)\backslash\mathrm{GF}(p)$:

  $8*\log_2(m)$ multiplications in $\mathrm{GF}(p)$

- $F(c,X)$ no roots in $\mathrm{GF}(p^2)[X]$: using Scipione del Ferro

  expected $2.4*\log_2(m)$ multiplications in $\mathrm{GF}(p)$

$F(c,X)*F(c^p,X) = (X^2 + G_0X + 1)(X^2 + G_1X + 1)(X^2 + G_2X + 1)$

with $G_i \in \mathrm{GF}(p^6)$, then

$P(c,X) = (X - G_0)(X - G_1)(X - G_2) \in \mathrm{GF}(p)[X]$,

$P(c,X) = X^3 + (c^p + c)X^2 + (c^{p+1} + c^p + c - 3)X + c^{2p} + c^2 + 2 - 2c^p - 2c$, and

$F(c,X)$ irreducible over $\mathrm{GF}(p^2)$ $\Leftrightarrow$ $P(c,X)$ irreducible over $\mathrm{GF}(p)$

Improved XTR parameter generation

Finding c such that $c = Tr(h)$ for h of order dividing $p^2 - p + 1$

$\Leftrightarrow$ $F(c,X)$ irreducible over $\mathrm{GF}(p^2)[X]$ $\Leftarrow$

- $Tr(h^{p+1}) \in \mathrm{GF}(p^2)\backslash\mathrm{GF}(p)$:

  $8*\log_2(m)$ multiplications in $\mathrm{GF}(p)$
- $F(c,X)$ no roots in $\mathrm{GF}(p^2)[X]$: using Scipione del Ferro

  expected $2.4*\log_2(m)$ multiplications in $\mathrm{GF}(p)$
- $X^3 + (c^p + c)X^2 + (c^{p+1} + c^p + c - 3)X + c^{2p} + c^2 + 2 - 2c^p - 2c \in \mathrm{GF}(p)[X]$

  no roots in $\mathrm{GF}(p)[X]$: using Scipione del Ferro

  expected $0.9*\log_2(m)$ multiplications in $\mathrm{GF}(p)$

- $c = (27\alpha^2 + 3\alpha)/19 \in \mathrm{GF}(p^2)$ or $c = (-27\alpha^2 - 24\alpha)/19 \in \mathrm{GF}(p^2)$

  if p is not 8 modulo 9:

  expected $0*\log_2(m)$ multiplications in $\mathrm{GF}(p)$

XTR parameter generation if p is not 8 modulo 9

If p is not 8 modulo 9:

$(Z^9 - 1)/(Z^3 - 1) = Z^6 + Z^3 + 1$ is irreducible over GF(p)

$\Rightarrow$ GF(p^6) $\approx$ GF(p)(ζ) with $\zeta^6 + \zeta^3 + 1 = 0$

$Q = (p^6 - 1)/(p^2 - p + 1)$, $a \in$ GF(p), $p \equiv 2 \bmod 9$,

$\Rightarrow$ trace over GF(p^2) of $(\zeta + a)^Q$ (of order dividing $p^2 - p + 1$)

equals $-3((a^2 - 1)^3\alpha + a^3(a^3 - 3a + 1)\alpha^2)/(a^6 - a^3 + 1) \in$ GF(p^2)

$a = 1/2$ results in $c = (27\alpha + 3\alpha^2)/19 \in$ GF(p^2)

$a = 2$ results in $c = (-27\alpha - 24\alpha^2)/19 \in$ GF(p^2)

XTR parameter generation if p is not 8 modulo 9

If p is not 8 modulo 9:

$$(Z^9 - 1)/(Z^3 - 1) = Z^6 + Z^3 + 1 \text{ is irreducible over } GF(p)$$

$\Rightarrow$ $GF(p^6) \approx GF(p)(\zeta)$ with $\zeta^6 + \zeta^3 + 1 = 0$

$Q = (p^6 - 1)/(p^2 - p + 1)$, $a \in GF(p)$, $p \equiv 5 \bmod 9$,

$\Rightarrow$ trace over $GF(p^2)$ of $(\zeta + a)^Q$ (of order dividing $p^2 - p + 1$)

equals $-3((a^2 - 1)^3\alpha^2 + a^3(a^3 - 3a + 1)\alpha)/(a^6 - a^3 + 1) \in GF(p^2)$

$a = 1/2$ results in $c = (27\alpha + 3\alpha^2)/19 \in GF(p^2)$

$a = 2$ results in $c = (-27\alpha - 24\alpha^2)/19 \in GF(p^2)$

XTR application example: Diffie–Hellman

given primes $p \equiv 2 \bmod 3$ and $q > 3$ with q dividing $p^2 - p + 1$, and $Tr(g)$ for g of order q

- A picks a, computes $Tr(g^a)$, sends it to B
- B receives $Tr(g^a)$, picks b, computes $Tr(g^b)$, sends it to A, and computes common key $Tr(g^{ab})$
- A receives $Tr(g^b)$, computes common key $Tr(g^{ab})$

XTR is secure, efficient, compact, easy to implement, with trivial parameter generation

Any disadvantages?

- Do we really trust GF(p^6)?
- Multiplication of $Tr(g^m)$ and $Tr(g^n)$ is non−trivial (but can usually be avoided)
- Signature verification is slow (just like other DL based schemes)
- Signature verification needs $Tr(g^k)$, $Tr(g^{k-1})$, $Tr(g^{k+1})$ (secret k)

But: $Tr(g^{k-1})$ follows from $Tr(g^k)$ and $Tr(g^{k+1})$
and $Tr(g^{k+1})$ can be computed quickly given $Tr(g^k)$