ATRODO / Net-Dropbear-0.14 / dropbear / libtomcrypt / changes

```
July 1st, 2018
v1.18.2
      -- Fix Side Channel Based ECDSA Key Extraction (CVE-2018-12437) (PR
      -- Fix potential stack overflow when DER flexi-decoding (CVE-2018-07
      -- Fix two-key 3DES (PR #390)
      -- Fix accelerated CTR mode (PR #359)
      -- Fix Fortuna PRNG (PR #363)
      -- Fix compilation on platforms where cc doesn't point to gcc (PR #3
      -- Fix using the wrong environment variable LT instead of LIBTOOL (P
      -- Fix build on platforms where the compiler provides __WCHAR_MAX__
      -- Fix & re-factor crypt_list_all_sizes() and crypt_list_all_constan
      -- Minor fixes (PR's #350 #351 #375 #377 #378 #379)

January 22nd, 2018
v1.18.1
      -- Fix wrong SHA3 blocksizes, thanks to Claus Fischer for reporting
      -- Fix NULL-pointer dereference in `ccm_memory()` with LTC_CLEAN_STA
      -- Fix `ccm_process()` being unable to process input buffers longer
      -- Fix the `register_all_{ciphers,hashes,prngs}()` return values (PR
      -- Fix some typos, warnings and duplicate prototypes in code & doc (
      -- Fix possible undefined behavior with LTC_PTHREAD (PR #337)
      -- Fix some DER bugs (PR #339)
      -- Fix CTR-mode when accelerator is used (OP-TEE/optee_os #2086)
      -- Fix installation procedure (Issue #340)

October 10th, 2017
v1.18.0
      -- Bugfix multi2
      -- Bugfix Noekeon
      -- Bugfix XTEA
      -- Bugfix rng_get_bytes() on windows where we could read from c:\dev
      -- Fixed the Bleichbacher Signature attack in PKCS#1 v1.5 EMSA, than
      -- Fixed a potential cache-based timing attack in CCM, thanks to Seb
      -- Fix GCM counter reuse and potential timing attacks in EAX, OCB an
         thanks to Raphaël Jamet
      -- Implement hardened RSA operations when CRT is used
      -- Enabled timing resistant calculations of ECC and RSA operations p
      -- Applied some patches from the OLPC project regarding PKCS#1 and p
         the hash algorithms from overflowing
      -- Larry Bugbee contributed the necessary stuff to more easily call
         from a dynamic language like Python, as shown in his pyTomCrypt
      -- Nikos Mavrogiannopoulos contributed RSA blinding and export of RS
         in OpenSSL/GnuTLS compatible format
      -- Patrick Pelletier contributed a smart volley of patches
      -- Christopher Brown contributed some patches and additions to ASN.1
      -- Pascal Brand of STMicroelectronics contributed patches regarding
         XTS mode and RSA private key operations with keys without CRT par
      -- RC2 now also works with smaller key-sizes
      -- Improved/extended several tests & demos
      -- Hardened DSA and RSA by testing (through Karel's perl-CryptX)
```

```
                -- Re-worked all makefiles
                -- Re-worked most PRNG's
                -- The code is now verified by a linter, thanks to Francois Perrad
                -- Documentation (crypt.pdf) is now built deterministically, thanks
                -- Add Adler32 and CRC32 checksum algorithms
                -- Add Base64-URL de-/encoding and some strict variants
                -- Add Blake2b & Blake2s (hash & mac), thanks to Kelvin Sherlock
                -- Add Camellia block cipher
                -- Add ChaCha (stream cipher), Poly1305 (mac), ChaCha20Poly1305 (enc
                -- Add constant-time mem-compare mem_neq()
                -- Add DER GeneralizedTime de-/encoding
                -- Add DSA and ECC key generation FIPS-186-4 compliance
                -- Add HKDF, thanks to RyanC (especially for also providing document
                -- Add OCBv3
                -- Add PKCS#1 v1.5 mode of SSL3.0
                -- Add PKCS#1 testvectors from RSA
                -- Add PKCS#8 & X.509 import for RSA keys
                -- Add stream cipher API
                -- Add SHA3 & SHAKE
                -- Add SHA512/256 and SHA512/224
                -- Add Triple-DES 2-key mode, thanks to Paul Howarth
                -- Brought back Diffie-Hellman

        May 12th, 2007
        v1.17 -- Cryptography Research Inc. contributed another small volley of pa
                 another to silence MSVC warnings.
                -- Added LTC_XCBC_PURE to XCBC mode which lets you use it in three-k
                -- [CRI] Added libtomcrypt.dsp for Visual C++ users.
                -- [CRI] Added more functions for manipulating the ECC fixed point c
                -- [CRI] Modified ecc_make_key() to always produce keys smaller than
                -- Elliptic Semiconductor contributed XTS chaining mode to the ciphe
                -- Fixed xcbc_init() keylen when using single key mode.
                -- Bruce Fortune pointed out a typo in the hmac_process() descriptio
                -- Added variable width counter support to CTR mode
                -- Fixed CMAC (aka OMAC) when using 64-bit block ciphers and LTC_FAS
                -- Fixed bug in ecc_is_valid() that would basically always return tr
                -- renamed a lot of macros to add the LTC_ prefix [e.g. RIJNDAEL =>

        December 16th, 2006
        v1.16 -- Brian Gladman pointed out that a recent change to GCM broke how t
                 so the code should be considered frozen now.
                -- Trevor from Cryptography Research Inc. submitted patches to conve
                 at runtime.
                -- Fixed various doxygen comments
                -- Added UTF8 support to the ASN1 code
                -- Fixed STOREXXH macros for x86 platforms (Fix found at Elliptic In
                -- Added makefile.unix which is BSD compatible, you have to manually
                -- removed a few lingering memcpy's
                -- Fixed memory free errors in ecc_sign_hash() that can arise if the
                -- Fixed incorrect return value in pkcs_1_pss_decode() which would c
                 would return CRYPT_OK to the caller
                -- ltc_ecc_mulmod() could leak memory if mp_init(&mu) failed, fixed.
                 bug?  Also fixed.  :-)
                -- Added Shamir's trick to the ECC side (defined as LTC_ECC_SHAMIR,
                -- Added Brian's vector #46 to the GCM code.  It catches the ctr cou
                 but they're not as easy to parse and I got a lot of other things
                -- Various other small fixes to the ECC code to clean up error handl
                 All of the errors were in cleaning up from heap failures.  So the
```

```
117
118
119         November 17th, 2006
120         v1.15 -- Andreas Lange found that if sha256_init DID fail in fortuna it wo
121                  Fortunately sha256_init cannot fail (as of v1.14) :-)
122               -- Andreas Lange contributed RMD-256 and RMD-320 code.
123               -- Removed mutex locks from fortuna_import as they create a deadlock
124               -- Added LTC_NO_PROTOTYPES to avoid prototyping functions like memse
125               -- David Eder caught a off by one overrun bug in pmac_done() which c
126                  smaller than the block size of the cipher, e.g. if you have a 4-b
127                  a 4-byte TAG it will store 4 bytes but return an outlen of 5.
128               -- Added signatures to the ECC and RSA benchmarks
129               -- Added LTC_PROFILE to run the PK tests only once in the timing dem
130               -- Andreas contributed PKCS #1 v1.5 code that merged cleanly with th
131                  (update: I had to fix it to include the digestInfo and what not.
132               -- Fixed a signed variable error in gcm_process() (hard to trigger b
133               -- Removed all memcmp/memset/memcpy from the source (replaced with X
134               -- Renamed macros HMAC/OMAC/PMAC to have a LTC_ prefix.  If you pass
135               -- Added XCBC-MAC support [RFC 3566]
136               -- fixed LOAD32H and LOAD64H to stop putting out that darn warning :
137               -- Added the Korean SEED block cipher [RFC 4269]
138               -- Added LTC_VALGRIND define which makes SOBER-128 and RC4 a pure PR
139                  Valgrind to debug your code (reported by Andreas Lange)
140               -- Made SOBER-128 more portable by removing the ASCII key in the tes
141               -- Martin Mocko pointed out that if you have no PRNGs defined the li
142                  hashes defined.
143               -- Sped up F8 mode with LTC_FAST
144               -- Made CTR mode RFC 3686 compliant (increment counter first), to en
145                  parameter you pass to ctr_start(), otherwise it will be LTC compl
146               -- Added ctr_test() to test CTR mode against RFC 3686
147               -- Added crypt_fsa() ... O_o
148               -- Fixed LTC_ECC_TIMING_RESISTANT so it once again builds properly (
149               -- Added ANSI X9.63 (sec 4.3.6) import/export of public keys (cannot
150                  hybrid compressed)
151               -- Added SECP curves for 112, 128, and 160 bits (only the 'r1' curve
152               -- Added 3GPP-F9 MAC (thanks to Greg Rose for the test vectors)
153               -- Added the KASUMI block cipher
154               -- Added F9/XCBC/OMAC callbacks to the cipher plugin
155               -- Added RSA PKCS #1 v1.5 signature/encrypt tests to rsa_test.c
156               -- Fix to yarrow_test() to not call yarrow_done() which is invalid i
157               -- Christophe Devine pointed out that Anubis would fail on various 6
158                  to mask it with 0xFF.  Thanks.  Fixed.
159
160         August 0x1E, 0x07D6
161         v1.14 -- Renamed the chaining mode macros from XXX to LTC_XXX_MODE.  Shoul
162               -- clean up of SHA-256
163               -- Chris Colman pointed out that der_decode_sequence_* allows LTC_AS
164                  Decoder [non-flexi decoder that is] is more strict now and requir
165               -- Steffen Jaeckel pointed out a typo in the user manual (re: rsa_ex
166                  nobody reads it.  :-)
167               -- Made GCM a bit more portable w.r.t. handling the CTR IV (e.g. & w
168               -- Add LTC_VERBOSE if you really want to see what test is doing :-)
169               -- Added SSE2 support to GCM [use GCM_TABLES_SSE2 to enable], shaves
170                  Shaved 4 cycles on a Prescott (Intel P4)
171                  Requires you align your gcm_state on a 16 byte boundary, see gcm_
172               -- Added missing prototype for f8_test_mode()
173               -- two fixes to CCM for corner cases [L+noncelen > 15] and fixing th
174               -- Franz Glasner pointed out the ARGTYPE=4 is not actually valid.  F
```

```
179
180         June 17th, 2006
181         v1.13 -- Fixed to fortuna_start() to clean up state if an error occurs.  N
182                  if I ever make fortuna pluggable
183               -- Mike Marin submitted a whole bunch of patches for fixing up the l
184               -- One of bugs found in the multi demo highlights that at least with
185                  they're unsigned long
186               -- Updated the FP ECC code to use affine points.  It's teh fast.
187               -- Made it so many functions which return CRYPT_BUFFER_OVERFLOW now
188                  do this (most do though).
189               -- Added F8 chaining mode.  It's super neato.
190
191         May 29th, 2006
192         v1.12 -- Fixed OID encoder/decoder/length to properly handle the first two
193               -- [Wesley Shields] Allows both GMP/LTM and TFM to be defined now.
194               -- [Wesley Shields] GMP pluggin is cleaner now and doesn't use depre
195               -- Added count_lsb_bits to get the number of leading LSB zero bits t
196               -- Fixed a bug in the INTEGER encoders for values of -(256**k)/2
197               -- Added BOOLEAN type to ASN.1 thingy-ma-do-hicky
198               -- Testprof doesn't strictly require GMP ... oops [Nils Durner]
199               -- Added LTC_CALL and LTC_EXPORT macros in tomcrypt_cfg.h to support
200                  (Thanks to John Kirk from Demonware)
201               -- In what has to be the best thing since sliced bread I bring you M
202                  ECC point multiplier.  It's fast, it's sexy and what's more it's
203                  You can tune it somewhat with FP_LUT (default to 8) for look-up w
204                  Read section 8.2 of the manual for more info.
205                  It is disabled by default, you'll have to build LTC with it defin
206               -- Fixed bug in ecc_test.c (from testprof) to include the 521 [not 5
207
208         April 4th, 2006
209         v1.11 -- Removed printf's from lrw_test ... whoops
210               -- lrw_process now checks the return of the cipher ecb encrypt/decry
211               -- lrw_start was not using num_rounds ...
212               -- Adam Miller reported a bug in the flexi decoder with elements pas
213               -- Bruce Guenter suggested I use --tag=CC for libtool builds where t
214               -- Optimized the ECC for TFM a bit by removing the useless "if" stat
215                  Actually shaved a good chunk of time off and made the code smalle
216                  will be totally omitted (ECC-256 make key times on a Prescott for
217               -- added missing CVS tags to ltc_ecc_mulmod.c
218               -- corrected typo in tomcrypt_cfg.h about what the file has been cal
219               -- corrected my address in the user manual.  A "bit" out of date.
220               -- added lrw_gen to tv_gen
221               -- added GMP plugin, only tested on a AMD64 and x86_32 Gentoo Linux
222               -- made testme.sh runs diff case insensitivityly [whatever...] cuz G
223               -- added LDFLAGS to the makefile to allow cross porting linking opti
224               -- added lrw_test() to the header file ... whoops
225               -- changed libtomcrypt.org to libtomcrypt.com .... mumble mumble
226               -- Updates to detect __STRICT_ANSI__ which is defined in --std=c99 m
227                  build LTC out of the box with c99 (note: it'll be slower as there
228               -- Updated pelican.c and aes_tab.c to undef tables not-required.  Th
229               -- Added LTC_NO_FAST to the makefile.icc to compensate for the fact
230
231         February 11th, 2006
232         v1.10 -- Free ecb/cbc/ctr/lrw structures in timing code by calling the "do
233               -- fixed bug in lrw_process() which would always use the slow update
234               -- vastly sped up gcm_gf_mult() when LTC_FAST is defined.  This spee
235               -- Removed NLS since there are some attacks against it.
236               -- fixed memory leak in rsa_import reported by John Kuhns
```

```
241               -- Added "easy button" define LTC_EASY and LTC will build with a sub
242                  configurations.  Tunable [see tomcrypt_custom.h]
243               -- Added some error detection to reg_algs() of the testprof.a librar
244               -- Similar fixes to timing demo (MD5 not defined when EASY is define
245               -- Added the NLS enc+mac stream cipher from QUALCOMM, disabled for t
246               -- Finally added an auto-update script for the makefiles.  So when I
247               -- Added LRW to the list of cipher modes supported
248               -- cleaned up ciphers definitions to remove cbc/cfb/ofb/ctr/etc from
249
250         November 24th, 2005
251         v1.08 -- Added SET and SET OF support to the ASN.1 side
252               -- Fixed up X macros, added QSORT to the mix [thanks SET/SETOF]
253               -- Added XMEMCMP to the list of X macros
254               -- In der_decode_sequence() the SHORT_INTEGER type was not being han
255               -- Fixed bug in math descriptors where if you hadn't defined MECC (E
256               -- Added RSA accelerators to the math descriptors to make it possibl
257               -- dsa_decrypt_key() was erroneously dependent on MECC not MDSA ...
258               -- Moved DSA size limits to tomcrypt_pk.h so they're defined with LT
259               -- cleaned up tomcrypt_custom.h to make customizable PK easier (and
260
261         November 18th, 2005
262         v1.07 -- Craig Schlenter pointed out the "encrypt" demo doesn't call ctr_s
263                  I added support to set the mode of the counter at init time
264               -- Fixed some "testprof" make issues
265               -- Added RSA keygen to the math descriptors
266               -- Fixed install_test target ... oops
267               -- made the "ranlib" program renamable useful for cross-compiling
268               -- Made the cipher accelerators return error codes.  :-)
269               -- Made CCM accept a pre-scheduled key to speed it up if you use the
270               -- Added "Katja" public key crypto.  It's based on the recent N = p^
271                  to it.  Note this code has been disabled not because it doesn't w
272                  analyzed.   It does carry some advantages over RSA (slightly smal
273                  some annoying "setup" issues like the primes are smaller which ma
274               -- Made makefile accept a NODOCS flag to disable the requirement of
275               -- Cleaned up rsa_export() since "zero" was handled with a SHORT_INT
276               -- Cleaned up the LIBTEST_S definitions in both GNU makefiles.  A fe
277               -- Made the cipher ecb encrypt/decrypt return an int as well, change
278               -- der_decode_choice() would fail to mark a NULL as "used" when deco
279               -- ecc_decrypt_key() now uses find_hash_oid() to clean up the code ;
280               -- Added mp_neg() to the math descriptors.
281               -- Swapped arguments for the pkcs_1_mgf1() function so the hash_idx
282               -- Made the math descriptors buildable when RSA has been undefined
283               -- ECC timing demo now capable of detecting which curves have been d
284               -- Refactored the ECC code so it's easier to maintain.  (note: the f
285               -- Updated the documentation w.r.t. ECC and the accelerators to keep
286               -- Fixed bug in ltc_init_multi() which would fail to free all alloca
287               -- Fixed bug in ecc_decrypt_key() which could possibly lead to overf
288               -- Added encrypt/decrypt to the DSA side (basically DH with DSA para
289               -- Updated makefiles to remove references to the old DH object files
290               -- ecc_import() now checks if the point it reads in lies on the curv
291               -- ECC code now ALWAYS uses the accelerator interface.  This allows
292                  ECC point add/dbl/mul code linked in.  Yeah space savings! Rah Ra
293               -- Added LTC_MUTEX_* support to Yarrow and Fortuna allowing you to u
294                  build time (e.g. LTC_PTHREAD == pthreads)
295               -- Added PPC32 support to the rotate macros (tested on an IBM PPC 40
296               -- Added ltc_mp checks in all *_make_key() and *_import() which will
297               -- the UTCTIME type was missing from der_length_sequence() [oops, oh
298               -- the main makefile allows you to rename the make command [e.g. MAK
```

```
303         v1.06 -- Fixed rand_prime() to accept negative inputs as a signal for BBS
304               -- Added fourth ARGCHK type which outputs to stderr and continues.
305               -- Removed the DH code from the tree
306               -- Made the ECC code fully public (you can access ecc_mulmod directl
307               -- Added ecc test to tv_gen
308               -- Added hmac callback to hash descriptors.
309               -- Fixed two doxy comment errors in the UTCTIME functions
310               -- rsa_import() can now read OpenSSL format DER public keys as well
311                  Note that rsa_export()  **ONLY** writes PKCS #1 formats
312               -- Changed MIN/MAX to only define if not already present.  -- Kirk J
313               -- Ported tv_gen to new framework (and yes, I made ecc vectors BEFOR
314               -- ported testing scripts to support pluggable math.  yipee!
315               -- Wrote a TFM descriptor ... yipee
316               -- Cleaned up LTC_FAST in CBC mode a bit
317               -- Merged in patches from Michael Brown for the sparc/sparc64 target
318               -- Added find_hash_oid() to search for a hash by its OID
319               -- Cleaned up a few stray CLEAN_STACKs that should have been LTC_CLE
320               -- Added timing resistant ECC, enable by defining LTC_ECC_TIMING_RES
321               -- Updated the ECC documentation as it was a bit out of date
322
323         June 27th, 2005
324         v1.05
325               -- Added Technote #6 which covers the current PK compliance.
326               -- Fixed buffer overflow in OAEP decoder
327               -- Added CHOICE to the list of ASN.1 types
328               -- Added UTCTIME to the list of ASN.1 types
329               -- Added MUTEX locks around descriptor table functions [but not on t
330                  All functions call *_is_valid() before using a descriptor index w
331                  it can be accessed.  However, during the operation [e.g. CCM] if
332                  undefined.
333               -- Minor updates to the manual to reflect recent changes
334               -- Added a catch to for an error that should never come up in rsa_ex
335
336         June 15th, 2005
337         v1.04
338               -- Fixed off by one [bit] error in dsa_make_key() it was too high by
339               -- ECC-224 curve was wrong [it was an ok curve just not NIST, so no
340               -- Removed point compression since it slows down ECC ops to save a m
341                  This makes the ecc export format incompatible with 1.03 [it shoul
342               -- Removed ECC-160 from timing and added the other curves
343
344         June 9th, 2005
345         v1.03
346               -- Users may want to note that on a P4/GCC3.4 platform "-fno-regmove
347               ------------------------------------------------------------------
348               -- Made it install the testing library in the icc/static makefiles
349               -- Found bug in ccm_memory.c which would fail to compile when LTC_CL
350               -- Simon Johnson proposed I do a fully automated test suite.  Hence
351               -- Added LTC_NO_TEST which forces test vectors off (regardless of wh
352               -- Added LTC_NO_TABLES which disables large tables (where possible,
353               -- New test script found a bug in twofish.c when TABLES was disabled
354               -- Added a LTC_FAST specific test to the testing software.
355               -- Updated test driver to actually halt on errors and just print the
356               -- Added bounds checking to Pelican MAC
357               -- Added BIT and OCTET STRING to the ASN.1 side of things.
358               -- Pekka Riikonen pointed out that my ctr_start() function should ac
359               -- Cleaned up warnings in testprof
360               -- Removed redundant mu and point mapping in ecc_verify_hash() so it
```

```
365               -- As part of the move for ECC to X9.62 I've changed the signature a
366               -- Pekka helped me clean up the PKCS #1 v2.1 [OAEP/PSS] code
367               -- Wrote new DER SEQUENCE coder/decoder
368               -- RSA, DSA and ECDSA now use the DER SEQUENCE code (saves a lot of
369               -- DSA output is now a DER SEQUENCE (so not compatible with previous
370               -- Added Technote #5 which shows how to build LTC on an AMD64 to hav
371               -- Changed temp variable in LOAD/STORE macros to "ulong32" for 32-bi
372               -- Added INSTALL_GROUP and INSTALL_USER which you can specify on the
373                  is to be installed as
374               -- Removed "testprof" from the default build.
375               -- Added IA5, NULL and Object Identifier to the list of ASN.1 DER su
376               -- The "no_oops" target (part of zipup) now scans for non-cvs files.
377               -- Added DERs for missing hashes, but just the OID not the PKCS #1 v
378               -- Removed PKCS #1 v1.5 from the tree since it's taking up space and
379               -- Kevin Kenny pointed out a few stray // comments
380               -- INTEGER code properly supports negatives and zero padding [Pekka!
381               -- Sorted asn1/der/ directory ... less of a mess now ;-)
382               -- Added PRINTABLE STRING type
383               -- Removed ECC-160 as it wasn't a standard curve
384               -- Made ecc_shared_secret() ANSI X9.63 compliant
385               -- Changed "printf" to "fprintf(stderr, " in the testbench... ;-)
386               -- Optimized the GCM table creation.  On 1KB packets [with key switc
387               -- Changed OID representation for hashes to be just a list of unsign
388               -- ECC code now uses Montgomery reduction ... it's even faster [ECC-
389               -- Added SHORT_INTEGER so users can easily store DER encoded INTEGER
390               -- Fixed OMAC code so that with LTC_FAST it doesn't require that LTC
391               -- ECC key export is now a simple [and documented] SEQUENCE, the "en
392               -- Thanks goes to the following testers
393                     Michael Brown             - Solaris 10/uSPARCII
394                     Richard Outerbridge       - MacOS
395                     Martin Carpenter          - Solaris 8/uSPARCII [Thanks for cle
396                     Greg Rose                 -  ... SunOS 5.8/SPARC [... what's w
397                     Matt Johnston             - MacOS X  [Thanks for pointing out
398
399         April 19th, 2005
400         v1.02
401               -- Added LTC_TEST support to gcm_test()
402               -- "pt/ct" can now be NULL in gcm_process() if you are processing ze
403               -- Optimized GCM by removing the "double copy" handling of the plain
404               -- Richard Outerbridge pointed out that x86_prof won't build on MACO
405                  erroneously refers to "mycrypt" all over the place.  Fixed.
406
407         April 17th, 2005
408         v1.01
409                ** Secure Science Corporation has supported this release cycle by s
410                   continuing support of this project has helped me maintain a stea
411                   stable and more efficient.
412                ---------------------------------------------------------------
413                -- Updated base64_decode.c so if there are more than 3 '=' signs it
414                -- Merged in latest mpi that fixed a few bugs here and there
415                -- Updated OAEP encoder/decoder to catch when the hash output is to
416                   Cleaned up PSS code too
417                -- Andy Bontoft fixed a bug in my demos/tests/makefile.msvc ... see
418                   afterall.  Thanks.
419                -- Made invalid ECC key sizes (configuration) not hard fault the pr
420                -- SAFER has been re-enabled after I was pointed to http://www.ciph
421                   [Mark Kotiaho]
422                -- Added CCM mode to the encauth list (now has EAX, OCB and CCM, c'
```

```
427                don't view this as a "huge issue" but it's just one less nit to
428             -- A new CVS has been setup on my Athlon64 box... if you want devel
429             -- Updated API for ECB and CBC shell code.  Now can process N whole
430             -- Introduced a new "hardware accel" framework that can be used to
431                calls.  Later on dependent code (e.g. OMAC, CCM) will be re-writ
432                if you [say] call ctr_encrypt() with a cipher descriptor that ha
433                be used (e.g. no code rewrites)
434             -- Now ships with 20% more love.
435             -- x86_prof now uses ECB shell code (hint: accelerators) and output
436                easier to compare hardware vs. software cipher implementations.
437             -- [Peter LaDow] fixed a typo w.r.t. XREALLOC macro (spelling count
438             -- Fixed bug with __x86_64__ where ROL64/ROR64 with LTC_NO_ROLC wou
439             -- Shipping with preliminary GCM code (disabled).  It's buggy (stac
440             -- Added Pelican MAC [it's an AES based fast MAC] to the list of su
441             -- Added LTC_FAST [and you can disable by defining LTC_NO_FAST] so
442                instead of one byte.  On my AMD64 this reduced the overhead for
443                that you either allow unaligned read/writes [e.g. x86_32/x86_64]
444                aligned access.  Only enabled for x86_* platforms by default sin
445             -- Added LTC_FAST support to PMAC (drops the cycle/byte by about 9
446             -- Updated "profiled" target to work with the new directory layout
447             -- Added [demo only] optimized RC5-CTR code to x86_prof demo to sho
448                [This has been removed prior to release... It may re-appear late
449             -- Added CCM acelerator callbacks to the list [now supports ECB, CT
450             -- Added chapter to manual about accelerators (you know you want it
451             -- Added "bswap" optimizations to x86 LOAD/STORE with big endian.
452             -- LTC_NO_ASM is now the official "disable all non-portable stuff"
453                disable any form of ASM and disable LTC_FAST load/stores.  Essen
454                trouble building the library (old GCCs for instance dislike the
455             -- Added tomcrypt_mac.h and moved MAC/encMAC functions from tomcryp
456             -- Added "done" function to ciphers and the five chaining modes [an
457             -- Changed install group to "wheel" from "root".
458             -- Replaced // comments with /**/ so it will build on older UNIX-li
459             -- x86_prof builds and runs with IntelCC fine now
460             -- Added "stest" build to intel CC to test static linked from withi
461             -- Moved testing/benchmark into testprof directory and build it as
462                testing info (hint: hardware developers ;-) )
463             -- Added CCM to tv_gen
464             -- Added demos to MSVC makefile
465             -- Removed -funroll-all-loops from GCC makefile and replaced with -
466             -- Fixed GCM prior to release and re-enabled it.  It has not been o
467             -- I've since optimized GCM and CCM.  They're close in speed but GC
468             -- For kicks I optimized the ECC code to use projective points.  Ge
469                speedup grows as the keysize grows.  Basically removing most pra
470             -- Added LTC_FAST support to OMAC/PMAC and doubled it's speed on my
471             -- Added GCM to tv_gen
472             -- Removed "makefile.cygwin_dll" as it's not really used by anyone
473             -- Updated a few files in the "misc" directory to have correct @fil
474             -- Removed "profile" target since it was slower anyways (go figure.
475
476      December 31st, 2004
477      v1.00
478             -- Added "r,s == 0" check to dsa_verify_hash()
479             -- Added "multi block" helpers for hash, hmac, pmac and omac routin
480                blocks of data with one call (added demos/multi.c to make sure t
481                -- Note these are not documented but they do have doxygen commen
482                -- Also I don't use them in other functions (like pkcs_5_2()) be
483             -- Added tweaked Anubis test vectors and made it default (undefined
484             -- Merged in fix for mp_prime_random_ex() to deal with MSB and LSB
```

```
489         December 23rd, 2004
490         v1.00rc1
491                -- Renamed "mycrypt_*" to "tomcrypt_*" to be more specific and prof
492                   Now just include "tomcrypt.h" instead of "mycrypt.h" to get LTC
493                -- Cleaned up makefiles to ensure all headers are correctly install
494                -- Added "rotate by constant" macros for portable, x86-32 and x86-6
495                   You can disable this new code with LTC_NO_ROLC which is useful f
496                -- Cleaned up detection of x86-64 so it works for ROL/ROR macros
497                -- Fixed rsa_import() so that it would detect multi-prime RSA keys
498                -- Sorted the source files by category and updated the makefiles ap
499                -- Added LTC_DER define so you can trim out DER code if not require
500                -- Fixed up RSA's decrypt functions changing "res" to "stat" to be
501                   with the signature variables nomenclature. (no code change just
502                -- Removed all labels starting with __ and replaced with LBL_ to av
503                -- Merged in LTM fix to mp_prime_random_ex() which zap'ed the most
504                   requested was a multiple of eight.
505                -- Made RSA_TIMING off by default as it's not terribly useful [and
506                -- Renamed SMALL_CODE, CLEAN_STACK and NO_FILE to have a LTC_ prefi
507                   with other programs.  e.g. SMALL_CODE => LTC_SMALL_CODE
508                -- Zed Shaw pointed out that on certain systems installing libs as
509                   is not root.  Now the makefiles allow this to be changed easily.
510                -- Renamed "struct _*_descriptor" to "struct ltc_*_descriptor" to a
511                   Also renamed _ARGCHK to LTC_ARGCHK
512                -- Zed Shaw pointed out that I still defined the prng structs in to
513                   weren't defined.  This made undef'ing FORTUNA break the build.
514                -- Added LTC_NO_ASM to disable inline asm macros [ROL/ROR/etc]
515                -- Changed RSA decrypt functions to change the output length variab
516                   it more consistent.
517                -- Added the 64-bit Khazad block cipher [NESSIE]
518                -- Added the 128-bit Anubis block cipher [with key support for 128.
519                -- Changes to several MAC functions to rename input arguments to mo
520                -- Removed FAST_PK support from dh_sys.c
521                -- Declared deskey() from des.c as static instead of a global
522                -- Added pretty much all practical GCC warning tests to the GCC [re
523                   warnings can easily be disabled for those with older copies of G
524                -- Added doxygen @ tags to the code...  phew that was a hell of a l
525                -- Also added pre-configured Doxygen script.
526                -- Cleaned up quite a few functions [ciphers, pk, etc] to make the
527                   E.g. ciphers keys are called "skey" consistently now.  The input
528                   These changes require no code changes on the behalf of developer
529                -- Started a SAFER+ optimizer [does encrypt only] which shaves a go
530                   at an expense of huge code.  It's in notes/etc/saferp_optimizer.
531                -- DSA sign/verify now uses DER encoded output/inputs and no LTC st
532                -- Matt Johnston found a missing semi-colon in mp_exptmod().  Fix h
533
534         October 29th, 2004
535         v0.99  -- Merged in the latest version of LTM which includes all of the re
536                -- Deprecated LTMSSE and removed it (to be replaced with TFM later
537                -- Stefan Arentz pointed out that mp_s_rmap should be extern
538                -- Kristian Gj?steen pointed out that there are typos in the
539                   "test" makefile and minor issues in Yarrow and Sober [just cosme
540                -- Matthew P. Cashdollar pointed out that "export" is a C++ keyword
541                   so changed the PRNG api to use "pexport" and "pimport"
542                -- Updated "hashsum" demo so it builds ;-)
543                -- Added automatic support for x86-64 (will configure for 64-bit li
544                -- Zhi Chen pointed out a bug in rsa_exptmod which would leak memor
545                -- Made hash functions "init" return an int.  slight change to API
546                -- Added "CHC" mode which turns any cipher into a hash the other LT
```

```
551               -- RSA is now fully joy.  rsa_export/rsa_import use PKCS #1 encodin
552                  compatible with other crypto libs that use the format.
553               -- Added support for x86-64 for the ROL/ROR macros
554               -- Changed the DLL and SO makefiles to optimize for speed, commente
555                  mycrypt_custom.h and added -DSMALL_CODE to the default makefile
556               -- Updated primality testing code so it does a minimum of 5 tests [
557                  (AFAIK not a security fix, just warm fuzzies)
558               -- Minor updates to the OMAC code (additional __ARGCHK and removed
559               -- Update build and configuration info which was really really real
560               ++ Minor update, switch RSA to use the PKCS style CRT
561
562       August 6th, 2004
563       v0.98  -- Update to hmac_init to free all allocated memory on error
564              -- Update to PRNG API to fix import/export functions of Fortuna and
565              -- Added test functions to PRNG api, RC4 now conforms ;-) [was a mi
566              -- Added the SOBER-128 PRNG based off of code donated by Greg Rose.
567              -- Added Tech Note #4 [notes/tech0004.txt]
568              -- Changed RC4 back [due to request].  It will now XOR the output s
569                 a stream cipher easily.
570              -- Update Fortuna's export() to emit a hash of each pool.  This mea
571                 entropy that was spread over all the pools isn't entirely lost w
572              -- Zhi Chen suggested a comment for rsa_encrypt_key() to let users
573                 PKCS #1 v2.0 padding.  (updated other rsa_* functions)
574              -- Cleaned up Noekeon to remove unrolling [wasn't required, was mes
575              -- Updated RC4 so that when you feed it >256 bytes of entropy it qu
576                 bytes.  Also removed the % from the key setup to speed it up a b
577              -- Added cipher/hash/prng tests to x86_prof to help catch bugs whil
578              -- Made the PRNG "done" return int, fixed sprng_done to not require
579              -- Spruced up mycrypt_custom.h to trap more errors and also help pr
580                 on non-i386 platforms by accident.
581              -- Added RSA/ECC/DH speed tests to x86_prof and cleaned it up to bu
582              -- Changed Fortuna to count only entropy [not the 2 byte header] ad
583                 reseed mechanism.
584              -- Added "export_size" member to prng_descriptor tables so you can
585                 the exported state for any given PRNG.
586              -- Ported over patch on LTM 0.30 [not ready to release LTM 0.31] th
587                 that used to result in negative zeroes when you multiplied zero
588                 (patch due to "Wolfgang Ehrhardt" <Wolfgang.Ehrhardt@munich.nets
589              -- Fixed rsa_*decrypt_key() and rsa_*verify_hash() to default to in
590                 if any of the higher level functions fail [before you get to the
591                 a known state].  Applied to both v2 and v1.5 padding helpers.
592              -- Added MACs to x86_prof
593              -- Fixed up "warnings" in x86_prof and tv_gen
594              -- Added a "profiled" target back [for GCC 3.4 and ICC v8].  Doesn'
595                 tinkering with.
596              -- Beefed up load/store test in demos/test
597
598              ++ New note, in order to use the optimized LOAD/STORE macros your p
599                 must support unaligned 32/64 bit load/stores.  The x86s support
600                 but some [ARM for instance] do not.  If your platform cannot per
601                 unaligned operations you must use the endian neutral code which
602                 any sort of platform.
603
604       July 23rd, 2004
605       v0.97b -- Added PKCS #1 v1.5 RSA encrypt/sign helpers (like rsa_sign_hash,
606              -- Added missing prng check to rsa_decrypt_key() [not critical as I
607                 descriptors directly in that function]
608              -- Merged in LTM-SSE, define LTMSSE before you build and you will g
```

```
                  -- Steven Dake <scd@broked.org> and Richard Amacker <ramacker@yahoo
                     fix pkcs_5_2().  It now matches the output of another crypto lib
                  -- Updated PRNG api.  Added Fortuna PRNG to the list of supported P
                  -- Fixed up the descriptor tables since globals are automatically z
                  -- Changed RC4 to store it's output.  If you want to encrypt with R
                     you'll have to do the XOR yourself.
                  -- Fixed buffer overflows/overruns in the HMAC code.

                  ++ API change for the PRNGs there now is a done() function per PRNG
                     should call it when you are done with a prng state.  So far it's
                     not absolutely required (won't cause problems) but is a good ide
                     start.


          June 23rd, 2004
          v0.97a ++ Fixed several potentially crippling bugs... [read on]
                  -- Fixed bug in OAEP decoder that would incorrectly report
                     buffer overflows. [Zhi Chen]
                  -- Fixed headers which had various C++ missing [extern "C"]'s
                  -- Added "extern" to sha384_desc descriptor which I removed by mist
                  -- Fixed bugs in ENDIAN_BIG macros using the wrong byte order [Matt
                  -- Updated tiger.c and des.c to not shadow "round" which is intrins
                     some C compilers.
                  -- Updated demos/test/rsa_test.c to test the RSA functionality bett
                  ++ This update has been tested with GCC [v3.3.3], ICC [v8] and MSVC
                     all on a x86 P4  [GCC/ICC tested in Gentoo Linux, MSVC in WinXP]
                  ++ Outcome: The bug Zhi Chen pointed out has been fixed.  So have t
                     that Matt Johnston found.

          June 19th, 2004
          v0.97  -- Removed spurious unused files [arrg!]
                  -- Patched buffer overflow in tim_exptmod()
                  -- Fixed buffer overrun bug in pkcs_1_v15_es_decode()
                  -- Reduced stack usage in PKCS #1 v2.0 padding functions (by severa
                  -- Removed useless extern's that were an artifact from the project
                  -- Replaced memcpy/memset with XMEMCPY and XMEMSET for greater flex
                  -- fixed bugs in hmac_done()/hmac_init()/[various others()] where I
                  -- Reduced stack usage in OMAC/PMAC/HMAC/EAX/OCB/PKCS#5 by mallocin
                     arrays (e.g. > 100 bytes or so).  Only in non-critical functions
                  -- "Zhi Chen" <zhi@massiveincorporated.com> pointed out that rsa_de
                     an incorrect output size (too large).  Fixed.
                  -- Added a "pretty" target to the GCC makefile.  Requires PERL.  It
                  -- Minor updates to ch1 of the manual.
                  -- Cleaned up the indentation and added comments to rsa_make_key(),
                     rsa_verify_hash()
                  -- Updated makefile.icc so the "install" target would work ;-)
                  -- Removed demos/test.c [deprecated from demos/test/test.c]
                  -- Changed MAXBLOCKSIZE from 128 to 64 to reflect the true size...

          May 30th, 2004
          v0.96  -- Removed GF and Keyring code
                  -- Extended OAEP decoder to distinguish better [and use a more unif
                  -- Changed PSS/OAEP API slightly to be more consistent with other P
                  -- rsa_exptmod() now pads with leading zeroes as per I2OSP.
                  -- added error checking to yarrow code
                  --  pointed out that tommath.h from this distro will overwrite tomm
                     from libtommath.  I changed this to ltc_tommath.h to avoid any s
                  -- Fixed bug in PSS encoder/decoder that didn't handle the MSB prop
```

```
675               -- replaced old test harness with new over-engineer'ed one in /demo
676               -- updated cbc/cfb/ofb/ctr code with setiv/getiv functions to chang
677               -- Added PKCS #1 v1.5 RSA encryption and signature padding routines
678               -- Added DER OID's to most hash descriptors (as many as I could fin
679               -- modded rsa_exptmod() to use timing-resilient tim_exptmod() when
680                  added #define RSA_TIMING which can turn on/off this feature.
681               -- No more config.pl so please just read mycrypt_custom.h for build
682               -- Small update to rand_prime()
683               -- Updated sha1, md5 and sha256 so they are smaller when SMALL_CODE
684                  you're going to have to undefine SMALL_CODE ;-)
685               -- Worked over AES so that it's even smaller now [in both modes].
686
687       May 12th, 2004
688       v0.95  -- Optimized AES and WHIRLPOOL for SMALL_CODE by taking advantage o
689                 the transforms are circulant.  AES dropped 5KB and WHIRLPOOL dro
690                 using the default build options on the x86.
691               -- Updated eax so the eax_done() would clear the state [like hmac,p
692                  CLEAN_STACK has been defined.
693               -- added LTC_TEST support to rmd160
694               -- updates to mycrypt_pk.h
695               -- updated rand_prime() to faciliate making RSA composites
696               -- DSA/RSA now makes composites of the exact size desired.
697               -- Refactored quite a bit of the code, fewer functions per C file
698               -- cleaned up the makefiles to organize the objects logically
699               -- added ICC makefile along with "profiled" targets for both GNU an
700               -- Marked functions for removal before v1.00 see PLAN for more info
701               -- GCC 3.4.0 tested and seems to work
702               -- Added PKCS #5 support
703               -- Fixed typo in comment header of .C files  ;-)
704               -- Added PKCS #1 OAEP and PSS support.
705
706       Feb 20th, 2004
707       v0.94  -- removed unused variables from ocb.c and fixed it to match known
708               -- Added PMAC support, minor changes to OMAC/EAX code [I think....]
709               -- Teamed up with Brian Gladman.  His code verifies against my vect
710                  verifies against his test vectors.  Hazaa for co-operation!
711               -- Various small changes (added missing ARGCHKs and cleaned up inde
712               -- Optimization to base64, removed unused variable "c"
713               -- Added base64 gen to demos/tv_gen.c
714               -- Fix to demos/x86_prof.c to correctly identify the i386 architect
715               -- Fixed up all of the PK code by adding missing error checking, re
716                  shrunk some stack variables, removed non-required stack variable
717                  error conversion from MPI to LTC codes.  I also spotted a few "o
718                  checking which could have been used to force the code to read pa
719                  the buffer (in theory, haven't checked if it would work) by a fe
720               -- Added checks to OUTPUT_BIGNUM so the *_export() functions cannot
721                  also modded it so it stores in the output provided to the functi
722                  the local stack) which saves memory and time.
723               -- Made SAFER default to disabled for now (plans are to cleanhouse
724               -- Added the 512-bit one-way hash WHIRLPOOL which clocks in at 138
725                  Athlon XP [for comparison, SHA-512 clocks in at 77 cycles per by
726                  teams new sbox design (not the original NESSIE one).
727
728
729       Jan 25th, 2004
730       v0.93  -- [note: deleted v0.93 changes by accident... recreating from memo
731               -- Fix to RC2 to not deference pointer before ARGCHK
732               -- Fix to NOEKEON to match published test vectors as well as cleane
```

```
737               -- Fix to DSA to check return of a few LTM functions I forgot [mp_t
738               -- Added common headers to all C files
739               -- CTR mode supports big and little [default] endian counters now.
740               -- fix to find_cipher_any() so that it can handle a fragmented ciph
741               -- added find_hash_any() akin to find_cipher_any().
742               -- Added EAX code to demos/tv_gen.c  Hazaa!
743               -- Removed SONY defines and files from codebase.
744               -- Added OCB support [patents be damned] and to demos/tv_gen.c
745               -- Merge all of the INPUT/OUTPUT BIGNUM macros (less toc) into mycr
746               -- Made appropriate changes to the debug string in crypt.c
747
748       Dec 24th, 2003
749       v0.92  -- Updated the config.pl script so the options have more details.
750               -- Updated demos/tv_gen to include RIPEMD hashes
751               -- Updated Twofish so when TWOFISH_ALL_TABLES is defined a pre-comp
752                  is included [speedup: slight, about 4k cycles on my Athlon].
753               -- Re-wrote the twofish large key generation [the four 8x32 key dep
754                  With both optimizations [e.g. TWOFISH_ALL_TABLES defined] a 128-
755                  in 26,000 cycles on my Athlon XP [as opposed to 49,000 before] w
756               -- config.pl has been updated so rmd128.o and rmd160.o are objects
757               -- Andrew Mann found a bug in rsa_exptmod() which wouldn't indicate
758                  (e.g. not PK_PRIVATE or PK_PUBLIC)
759               -- Fixed up demos/x86_prof so it sorts the output now :-)
760               -- The project is now powered by radioactive rubber pants.
761               -- Fixed dh_encrypt_key() so if you pass it a hash with a smaller o
762                  will return CRYPT_INVALID_HASH [to match what ecc_encrypt_key()
763               -- Merge the store/encrypt key part of ecc_encrypt_key() as per dh_
764               -- Massive updates to the prime generation code.  I use the LTM ran
765                  interface between the LTC PRNG's and the LTM generic prng protot
766                  depending on the input size.  This nicely speeds up most prime g
767               -- Added SHA-224 to the list of hashes.
768               -- Made HMAC test vectors constant and static [takes ROM space inst
769               -- This release was brought to you by the letter P which stands for
770               -- Added generic HASH_PROCESS macro to mycrypt_hash.h which simplif
771                  I also optimized the compression functions of all but MD2 to not
772               -- Removed the division from the Blowfish setup function [dropped 3
773               -- Added stack cleaning to rijndael, cast5 so now all ciphers have
774               -- Added Skipjack to the list of ciphers [made appropriate changes
775                  demos/x86_prof.c]
776               -- Added mechanical testing to cipher test vector routines.  Now it
777                  compares.  Any fault (e.g. bug in code, compiler) in the routine
778                  stress test the key gen though...
779               -- Matt Johnson found a bug in the blowfish.c  apparently I was out
780                  The code now builds with any config.  Thanks.
781               -- Added OMAC1 Message Authentication Code support to the library.
782               -- Re-prototyped the hash "process" and "done" to prevent buffer ov
783                  Updated HMAC code to use them too.  Hazaa!
784               -- Fixed bug in ECC code which wouldn't do an _ARGCHK on stat in ec
785               -- Fixed [temp fix] bug in all PK where the OUTPUT_BIGNUM macros wo
786                  conversion [now returns CRYPT_MEM, will fix it up better later]
787               -- Added DSA to the list of supported PK algorithms.
788               -- Fixed up various ciphers to &255 the input key bytes where requi
789                  problems on platforms where CHAR_BIT != 8
790               -- Merged in LibTomMath v0.28
791               -- Updated demos/x86_prof.c to use Yarrow during the key sched test
792                  /dev/random].
793               -- Added OMAC/HMAC tests to demos/tv_gen and I now store the output
794               -- Fixed a bug in config.pl that wouldn't have TWOFISH_TABLES defin
```

```
                -- Updated notes/tech0003.txt to take into account the existence of
                -- Slight changes to Noekeon, with SMALL_CODE undefined it uses a f
                   on my Athlon (35 cycles per byte or 410.4Mbit/sec at 1795Mhz)
                -- Added _ARGCHK() calls to is_prime() for the two input pointers.

        Sept 25th, 2003
        v0.91   -- HMAC fix of 0.90 was incorrect for keys larger than the block si
                -- Added error CRYPT_FILE_NOTFOUND for the file [hmac/hash] routine
                -- Added RIPEMD hashes to the hashsum demo.
                -- Added hashsum demo to MSVC makefile.
                -- Added RMD160 to the x86_prof demo [oops]
                -- Merged in LibTomMath-0.27 with a patch to mp_shrink() that will
                   Fixes another potential memory leak.

        Sept 7th, 2003
        v0.90   -- new ROL/ROR for x86 GCC
                -- Jochen Katz submitted a patch to the makefile to prevent "make"
                   when not required.
                == By default the KR code is not enabled [it's only a demo anyways!
                -- changed the "buf" in ecc_make_key from 4KB to 128 bytes [since t
                -- hmac_done() now requires you pass it the size of the destination
                   buffer overflows.  (API CHANGE)
                -- hmac/hash filebased routines now return CRYPT_NOP if NO_FILE is
                -- I've removed the primes from dh.c and replaced them with DR safe
                   configuration of LibTomMath.  Check out these comparisons on a 1

        768-bit,     4 vs.  10
        1024-bit,    8 vs.  18
        1280-bit,   12 vs.  34
        1536-bit,   20 vs.  56
        1792-bit    28 vs.  88
        2048-bit,   40 vs. 124
        2560-bit,   71 vs. 234
        3072-bit,  113 vs. 386
        4096-bit,  283 vs. 916

                   Times are all in milliseconds for key generation.  New primes ti
                   incompatible with previous releases.  However, this addition is
                   reductions for quite some time.
                -- Added RIPE-MD 128 and 160 to the list of supported hashes [10 in
                -- The project has been released as public domain.  TDCAL no longer

        July 15th, 2003
        v0.89   -- Fix a bug in bits.c which would prevent it from building with ms
                -- Merged in LibTomMath v0.24 [and I used the alloc/free macros thi
                -- Removed the LTC version of next_prime() and replaced it with a c
                   mp_prime_next_prime() from LibTomMath
                -- reverted bits.c to the 0.86 copy since the new one doesn't build
                   or cygwin.

        Jul 10th, 2003
        v0.88   -- Sped up CAST5 key schedule for MSVC
                -- added "ulong32" which allows people on 64-bit platforms to force
                   ciphers like blowfish and AES to be 32-bits.  E.g. when unsigned
                -- Optimized the SAFER-SK64, SAFER-SK128, SAFER+, RC5 and RC6 key s
                -- Optimized SHA-1 and SHA-256 quite a bit too.
                -- Fixed up the makefile to use -fomit-frame-pointer more liberally
                -- Added tv_gen program which makes test vectors for ciphers/hashes
```

```
861               -- Improved the AES and Twofish key schedule [faster, more constant
862               -- Tons of optimizations here and there.
863
864       Jun 15th, 2003
865       v0.86  -- Fixed up AES to workaround MSVC optimizer bug
866               -- Merged in fresh LTM base [based on v0.20] so there are no warnin
867               -- Wrote x86_prof which will time the hashes and ciphers downto cyc
868               -- Fixed up demos/encrypt to remove serpent_desc from the list
869               -- Re-enabled MSVC optimizations w00t w00t
870               -- Replaced "errno" with "err" in all functions that had it so it w
871                  with the global "errno"
872               -- Removed a set of unused variables from certain functions
873               -- Removed {#line 0 "..."} stuff from mpi.c to comply with ISO C  :
874
875       Jun 11th, 2003
876       v0.85  -- Swapped in a new AES routine
877               -- Removed Serpent
878               -- Added TDCAL policy document
879
880       Jun 1st, 2003
881       v0.84  -- Removed a 4KB buffer from rsa_decrypt_key that wasn't being used
882               -- Fixed another potential buffer problem.  Not an overflow but cou
883                  PK import routines to read past the end of the buffer.
884               -- Optimized the ECC mulmod more by removing a if condition that wi
885               -- Optimized prime.c to not include a 2nd prime table, removed code
886                  test from LibTomMath now
887               -- Added LTC_TEST define which when defined will enable the test ve
888               -- Removed ampi.o from the depends cuz it ain't no not working in *
889
890
891       Mar 29th, 2003
892       v0.83  -- Optimized the ecc_mulmod, it's faster and takes less heap/stack
893               -- Fixed a free memory error in ecc_mulmod and del_point which woul
894               -- Fixed two serious bugs in rsa_decrypt_key and rsa_verify_hash th
895                  buffer overflow.
896               -- Fixed a bug in the hmac testing code if you don't register all t
897                  errors now.
898
899       Mar 15th, 2003
900       v0.82  -- Manual updated
901               -- Added MSVC makefile [back, actually its written from scratch to
902               -- Change to HMAC helper functions API to avoid buffer overflow [so
903               -- the rsa_encrypt_key was supposed to reject key sizes out of boun
904                  same fix to the rsa_sign_hash
905               -- Added code to ensure that that chaining mode code (cfb/ofb/ctr/c
906                  structures when being called.  E.g. the indexes to the pad/ivs a
907               -- Cleaned up the DES code and simplified the core desfunc routine.
908               -- Simplified one of the boolean functions in MD4
909
910       Jan 16th, 2003
911       v0.81  -- Merged in new makefile from Clay Culver and Mike Frysinger
912               -- Sped up the ECC mulmod() routine by making the word size adapt t
913                  operations on 521-bit keys now (translates to about 8ms on my At
914                  as much as possible.  This sped the routine up quite a bit.
915               -- Fixed a huge flaw in ecc_verify_hash() where it would return CRY
916               -- Fixed up config.pl by fixing an invalid query and the file is sa
917                  (fix due to Mika Bostr?m)
918               -- Merged in LibTomMath for kicks
```

```
923         Dec 16th, 2002
924         v0.80  -- Found a change I made to the MPI that is questionable.  Not quit
925                   with the digit shifting.  In v0.79 I simply truncated without ze
926                   testing but I fixed it up none the less.
927                -- Optimized s_mp_mul_dig() from MPI to do a minimal number of pass
928                -- Fixed in rsa_exptmod() where I was getting the size of the resul
929                   but the fixed code is more readable.
930                -- Fixed slight bug in dh_sign_hash() where the random "k" value wa
931                   also made the #define FAST_PK speed up signatures as well.  Esse
932                   limit any private exponent to 256-bits.   Note that when FAST_PK
933                   binary or source incompatible with a copy of the library with it
934                -- Removed the DSA code.  If you want fast diffie-hellman just defi
935                -- Updated dh_sign_hash()/dh_verify_hash() to export "unsigned" big
936                   compatible with the previous release... sorry!  I've performed t
937                -- Fixed up the PK code to remove all use of mp_toraw() and mp_read
938                -- Fixed a bug in the DH code where it missed trapping a few errors
939                -- Fixed a slight "its-not-a-bug-but-could-be-done-better" bug in t
940                   testing to ensure that in the loop that searches for the next ca
941                   65000.  Should have been testing for MP_DIGIT_MAX
942                -- Spruced up the config.pl script.  It now makes a header file "my
943                   you include mycrypt.h.  This allows you to add libtomcrypt to a
944                   system around.  Note that you should use the makefile it writes
945                -- Used splint to check alot of the code out.  Tons of minor fixes
946                -- Also made all the internal functions of MPI are now static to av
947                -- **Notice**:  There are no planned future releases for at least a
948
949         Dec 14th, 2002
950         v0.79  -- Change to PK code [binary and source].  I made it so you have to
951                   *_verify_hash functions.  This prevents malformed packets from p
952                   the packet header size [by 4 bytes].
953                -- Made the test program halt on the first error it occurs.  Also m
954                -- Wrote the first chapter of my new book [DRAFT!], not in this pac
955                -- Included a perl script "config.pl" that will make "makefile.out"
956                -- Added shell script to look for latest release
957                -- Merge DH and ECC key defines from mycrypt_cfg.h into the makefil
958                -- updated the makefile to use BSD friendly archiving invokations
959                -- Changed the DH and ECC code to use base64 static key settings [e
960                   and is ever-so-slightly faster than before.
961                -- added "mp_shrink" function to shrink the size of bignums.  Speci
962                -- Added new exptmod function that calculates a^b mod c with fewer
963                   sized numbers].  Also added a "low mem" variant that doesn't use
964                   heap todo the calculation.  Both are #define'able controlled
965                -- Added XREALLOC macro to provide realloc() functionality.
966                -- Added fix where in rsa_import() if you imported a public key or
967                   not being used.
968                -- Fixed potential bug in the ECC code.  Only would occur on platfo
969                   often!]
970                -- Fixed up the ECC point multiplication, its about 15% faster now
971                -- While I was at it [since the lib isn't binary backwards compatib
972                   so they export as "unsigned" types saving 1 byte per bignum outp
973
974         Nov 28th, 2002
975         v0.78  -- Made the default ARGCHK macro a function call instead which redu
976                -- Fixed a bug in the XTEA keysize function which called ARGCHK inc
977                -- Added Noekeon block cipher at 2,800 bytes of object code and 345
978                -- Made the KR code check if the other PK systems are included [pro
979                -- Made "aes" an alias for Rijndael via a pre-processor macro.  Now
980                   Thanks to Jean-Luc Cooke for the "buzzword conformance" suggesti
```

```
                the code is no longer source compatible but still binary compati
             -- Fixed a few other minor bugs in the PK import code while I was a

Nov 26th, 2002
v0.77  -- Updated the XTEA code to use pre-computed keys.  With optimizati
          compared to the 121Mbit/sec before.  It is 288 bytes bigger than
       -- Cleaned up some of the ciphers and hashes (coding style, cosmeti
       -- Optimized AES slightly for 256-bit keys [only one if statement n
       -- Removed most test cases from Blowfish, left three of them there.
       -- Changed the primality routines around.  I now use 8 rounds of Ra
          step and the "rand_prime" function uses a modified sieve that av
       -- Fixed a bug in the ECC/DH signatures where the keys "setting" va
          that a invalid value could have caused segfaults, etc...
       -- **NOTE** Changed the way the ECC/DH export/import functions work
          with v0.76.  Essentially insteading of exporting the setting ind
          if you ever re-configure which key settings are supported the li
          keys.
       -- Optimized Blowfish by inlining the round function, unrolling it
          rest.  It achieves a rate of 425Mbit/sec with the new code compa
          object file is 7,813 bytes compared to 8,663 before and is 850 b
          faster!
       -- Optimized Twofish as well by inlining the round function.  Gets
          and the code is only 78 bytes larger than the previous copy.
       -- Removed SMALL_PRIME_TAB build option.  I use the smaller table a
       -- Fixed some mistakes concerning prime generation in the manual.
       -- [Note: sizes/speeds are for GCC 3.2 on an x86 Athlon XP @ 1.53Gh

Nov 25th, 2002
v0.76  -- Updated makefiles a bit more, use "-Os" instead of "-O2" to opti
          downto 265KB using GCC 3.2 on my x86 box.
       -- Updated the SAFER+, Twofish and Rijndael test vector routine to
       -- Updated all other test vector routines to return as soon as an e
       -- fixed a bug in the test program where errors in the hash test ro
          correctly.  I found this by temporarily changing one of the byte
          hashes check out [the demos/test.c would still have reported an


Nov 24th, 2002
v0.75  -- Fixed a flaw in hash_filehandle, it should ARGCHK that the fileh
       -- Fixed a bug where in hash_file if the call to hash_filehandle fa
          not be closed.
       -- Added more strict rules to build process, starting to weed out "
          In the next release "-Wconversion" will be enabled which will de

Nov 22nd, 2002 [later in the day]
v0.74  -- Wrote a small variant of SAFER+ which shaved 50KB off the size o
       -- Wrote a build option to remove the PK packet functions [keeps th
       -- Wrote a small variant of Rijndael (trimmed 13KB)
       -- Trimmed the TIGER/192 hash function a bit
       -- Overall the entire lib compiled is 295KB [down from 400KB before
       -- Fixed a few minor oversights in the MSVC makefile

Nov 22nd, 2002
v0.73  -- Fixed bug in RC4 code where it could only use 255 byte keys.
       -- Fixed bug in yarrow code where it would allow cast5 or md2 to be
       -- Removed the ecc compress/expand points from the global scope.  R
       -- Fixed bug where if you used the SPRNG you couldn't pass NULL as
          able todo since the SPRNG has no state...
```

```
1047
1048        Nov 21th, 2002
1049        v0.72  -- Fixed bug in the prime testing.  In the Miller-Rabin test I was
1050                  The math still worked out fine because in effect it was performi
1051                  works properly
1052               -- Fixed some of the code where it was still using the old error sy
1053               -- Sped up the RSA decrypt/sign routines
1054               -- Optimized the ecc_shared_secret routine to not use so much stack
1055               -- Fixed up the makefile to make releases where the version # is in
1056                  to
1057
1058        Nov 19th, 2002
1059        v0.71  -- HELP TOM.  I need tuition for the January semester.  Now I don't t
1060                  but I really need the help!  See my website http://tom.iahu.ca/h
1061                  if you can!
1062               ------------------------------------------------------------------
1063               -- Officially the library is no longer supported in GCC 3.2 in wind
1064                  In windows you can either use GCC 2.95.3 or try your luck with 3
1065                  "-fomit-frame-pointer" is broken in the windows build [but not t
1066                  If you simply must use 3.2 then I suggest you limit the optimiza
1067               -- Started new error handling API.  Similar to the previous except
1068                  CRYPT_ERROR
1069               -- Added my implementation of the MD2 hash function [despite the er
1070               -- Merged in more changes from Sky Schulz.  I have to make mention
1071                  getting me motivated to make some much needed updates to the lib
1072               -- Fixed one of the many mistakes in the manual as pointed out by D
1073               -- Fixed a bug in the RC4 code [wasn't setting up the key correctly
1074               -- Added my implementation of the CAST5 [aka CAST-128] block cipher
1075               -- Fixed numerous bugs in the PK code.  Essentially I was "freeing"
1076                  required nor a good a idea [double free].
1077               -- Tom needs a job.
1078               -- Fixed up the test harness as requested by Sky Schulz.  Also modi
1079                  and count # of ops performed.  This is more suitable than say en
1080                  where it could take minutes!
1081               -- Modified test programs hashsum/encrypt to use the new algorithms
1082               -- Removed the PKCS code since it was incomplete.  In the future I
1083                  provides PKCS support...
1084               -- updated the config system so the #defines are in the makefiles i
1085               -- Willing to work on an hourly basis for 15$ CDN per hour.
1086               -- updated the test program to not test ciphers not included
1087               -- updated the makefile to make "rsa_sys.c" a dependency of rsa.o [
1088               -- fixed numerous failures to detect buffer overflows [minor] in th
1089               -- fixed the safer [64-bit block version] test routines which didn'
1090                  function
1091               -- check out my CV at http://tom.iahu.ca/cv.html
1092               -- removed the GBA makefile and code from demos/test.c [not a parti
1093               -- merged in rudimentary [for testing] PS2 RNG from Sky Schulz
1094               -- merged in PS2 timer code [only shell included due to NDA reasons
1095               -- updated HMAC code to return errors where possible
1096               -- Thanks go to Sky Schulz who bought me a RegCode for TextPad [the
1097
1098        Nov 12th, 2002
1099        v0.70  -- Updated so you can swap out the default malloc/calloc/free routi
1100               -- Sky Schulz contributed some code towards autodetecting the PS2 i
1101               -- Added PS2 makefile contributed by Sky Schulz [see a pattern form
1102               -- Added ability to have no FILE I/O functions at all (see makefile
1103               -- Added support for substituting out the clock() function (Sky Sch
1104               -- Fixed up makefile to include new headers in the HEADERS variable
```

```
1109                too well as a block cipher.
1110             -- Fixed ARGCHK macro usage when ARGTYPE=1 throughout the code
1111             -- updated makefile to make subdirectory properly (Sku Schulz)
1112             -- Started towards new API setup.  Instead of checking for "== CRYP
1113                In future releases functions will return things other than CRYPT
1114                thread safe error reporting.  The manual will be updated to refl
1115                errors are returned as CRYPT_ERROR (except as noted) but in futu
1116             -- Removed the zlib branch since its not really required anyways.
1117
1118       Nov 11th, 2002
1119       v0.69  -- Added ARGCHK (see mycrypt_argchk.h) "arguement checking" to all
1120             -- Note I forgot to change the CRYPT version tag in v0.68... fixed
1121
1122       Nov 8th, 2002
1123       v0.68  -- Fixed flaw in kr_import/kr_export that wasted 4 bytes.  Source b
1124             -- Fixed bug in kr_find_name that used memcmp to match strings.  Us
1125             -- kr_clear now sets the pointer to NULL to facilate debugging [e.g
1126             -- static functions in _write/_read in keyring.c now check the retu
1127             -- Updated blowfish/rc2/rc5/rc6 keysize() function to not reject ke
1128                respective ciphers can use.
1129             -- Fixed a bug in hashsum demo that would report the hash for files
1130
1131       Oct 16th, 2002
1132       v0.67  -- Moved the function prototypes into files mycrypt_*.h.  To "insta
1133                header files "*.h" from the base of this project into your globa
1134             -- Made the OFB/CFB/CTR functions use "unsigned long" for the lengt
1135             -- Added keyring support for the PK functions
1136             -- ***API CHANGE*** changed the ecc_make_key and dh_make_key to act
1137                move the first argument to the next to last.
1138             -- Fixed bug in dh_test() that wouldn't test the primality of the o
1139             -- replaced the primes in the DH code with new ones that are larger
1140                associated with.  That is a 1024-bit DH key will have a 1025-bit
1141             -- cleaned up all the PK code, changed a bit of the API around [not
1142             -- major editing of the manual, started Docer program
1143             -- added 160 and 224 bit key settings for ECC.  This makes the DH a
1144             -- Added an additional check for memory errors in is_prime() and cl
1145             -- Removed ID_TAG from all files [meh, not a big fan...]
1146             -- Removed unused variable from yarrow state and made AES/SHA256 th
1147             -- Fixed a bug in the Yarrow code that called prng_is_valid instead
1148             -- The ECB/CBC/OFB/CFB/CTR wrappers now check that the cipher is va
1149                Returns int now instead of void.
1150
1151       Sept 24th, 2002
1152       v0.66  -- Updated the /demos/test.c program to time the hashes correctly.
1153                tests meaning its possible to run on RNG less platforms
1154             -- Updated the /demos/hashsum.c program to hash from the standard i
1155             -- Updated the RSA code to make keys a bit quicker [update by Wayne
1156                time.
1157             -- Dan Kaminsky suggested some cleanups for the code and the MPI co
1158                Code ships in unix LF format by default now too... will still bu
1159                to read the stuff you'll have to convert it
1160             -- Changes to the manual to reflect new API [e.g. hash_memory/file
1161
1162       Sept 20th, 2002
1163       v0.65  -- Wayne Scott (wscott@bitmover.com) made a few of suggestions to i
1164                importantly he pointed out the math lib is not really required.
1165                different platforms.  According to him with only a few troubles
1166                library worked as it was supposed to.  You can find the list at
```

```
1171
1172        Sept 19th, 2002
1173        v0.64  -- wrote makefile for the GBA device [and hacked the demos/test.c f
1174               -- Fixed error in PK (e.g. ECC, RSA, DH) import functions where I w
1175               -- fixed more typos in the manual
1176               -- removed all unused variables from the core library (ignore the I
1177               -- added "const char *crypt_build_settings" string which is a build
1178                  of all the build time options.  Useful for debugging since you c
1179                  exactly you had set for the mycrypt_cfg.h file.
1180               -- Added control over endianess.  Out of the box it defaults to end
1181                  configure the library for your platform.  Using this I boosted R
1182                  Athlon box.  See "mycrypt_cfg.h" for more information.
1183
1184        Sept 11th, 2002
1185        v0.63  -- Made hashsum demo output like the original md5sum program
1186               -- Made additions to the examples in the manual (fixed them up a bu
1187               -- Merged in the base64 code from Wayne Scott (wscott@bitmover.com)
1188
1189        Aug 29th, 2002
1190        v0.62  -- Added the CLEAN_STACK functionality to several of the hashes I f
1191
1192        Aug 9th, 2002
1193        v0.61  -- Fixed a bug in the DES code [oops I read something wrong].
1194
1195        Aug 8th, 2002
1196        v0.60  -- Merged in DES code [and wrote 3DES-EDE code based on it] from Do
1197
1198        Aug 7th, 2002
1199        v0.59  -- Fixed a "unsigned long long" bug that caused v0.58 not to build
1200               -- Cleaned up a little in the makefile
1201               -- added code that times the hash functions too in the test program
1202
1203        Aug 3rd, 2002
1204        v0.58  -- Added more stack cleaning conditionals throughout the code.
1205               -- corrected some CLEAR_STACK conditionals... should have been CLEA
1206               -- Simplified the RSA, DH and ECC encrypt() routines where they use
1207                  now they only make one call to ctr_encrypt()/ctr_decrypt().
1208
1209        Aug 2nd, 2002
1210        v0.57  -- Fixed a few errors messages in the SAFER code to actually report
1211               -- rsa_encrypt() uses the "keysize()" method of the cipher being us
1212                  key size.  By default rsa_encrypt() will choose to use a 256-bit
1213                  down if required.
1214               -- The rsa_exptmod() function will now more reliably detect invalid
1215               -- The padding method for RSA is more clearly documented.  Namely i
1216                  N then your modulus must be of length 1+3N.  So to sign a messag
1217                  145 byte (1160 bits) modulus.  This is all in the manual now.
1218               -- Added build option CLEAN_STACK which will allow you to choose wh
1219                  cipher/hash call
1220               -- Sped up the hash "process()" functions by not copying one byte a
1221               ++ (added just after I uploaded...)
1222                  MD4 process() now handles input buffers > 64 bytes
1223
1224        Aug 1st, 2002
1225        v0.56  -- Cleaned up the comments in the Blowfish code.
1226               -- Oh yeah, in v0.55 I made all of the descriptor elements constant
1227               -- fixed a couple of places where descriptor indexes were tested wr
1228                  to mess up.
```

```
1233
1234        July 29th, 2002
1235        v0.55  -- My god stupid Blowfish has yet again been fixed.  I swear I hate
1236                  library.  Use AES or something else cuz I really hate Blowfish a
1237               -- Partial PKCS support [hint DONT USE IT YET CUZ ITS UNTESTED!]
1238
1239        July 19th, 2002
1240        v0.54  -- Blowfish now conforms to known test vectors.  Silly bad coding t
1241               -- RC5/RC6/Serpent all have more test vectors now [and they seemed
1242
1243        July 18th, 2002
1244        v0.53  -- Added more test vectors to the blowfish code just for kicks [and
1245               -- added prng/hash/cipher is_valid functions and used them in all o
1246                  with an invalid index ever now.
1247               -- Simplified the Yarrow code once again :-)
1248
1249        July 12th, 2002
1250        v0.52  -- Fixed a bug in MD4 where the hash descriptor ID was the same as
1251                  all the routines...
1252               -- Fixed the comments in SHA-512 to be a bit more meaningful
1253               -- In md4 I made the PADDING array const [again to store it in ROM]
1254               -- in hash_file I switched the constant "512" to "sizeof(buf)" to b
1255               -- in SHA-1's test routine I fixed the string literal to say SHA-1
1256               -- Fixed a logical error in the CTR code which would make it skip t
1257                  the CTR code from v0.52 will be incompatible [binary wise] with
1258                  sense this way.
1259               -- Added {} braces for as many if/for/blocks of code I could find.
1260                  must have {} braces around it.
1261               -- made the rounds table in saferp_setup const [again for the ROM t
1262               -- fixed RC5 since it no longer requires rc5 to be registered in th
1263                  be part of the table...
1264               -- the packet.c code now makes crypt_error literal string errors wh
1265               -- cleaned up the SAFER+ key schedule to be a bit easier to read.
1266               -- fixed a huge bug in Twofish with the TWOFISH_SMALL define.  Beca
1267                  changed the "g_func()" to be called indirectly.  I forgot to act
1268                  g_func() function which caused it not to work... [does now :-)]
1269
1270        July 11th, 2002
1271        v0.51  -- Fixed a bug in SHA512/384 code for multi-block messages.
1272               -- Added more test vectors to the SHA384/512 and TIGER hash functio
1273               -- cleaned up the hash done routines to make more sense
1274
1275        July 10th, 2002
1276        v0.50  -- Fixed yarrow.c so that the cipher/hash used would be registered.
1277                  a bug where the SAFER+ name was "safer" but should have been "sa
1278               -- Added an element to the hash descriptors that gives the size of
1279               -- Cleaned up the support for HMAC's
1280               -- Cleaned up the test vector routines to make the test vector data
1281                  placed in ROM not RAM now.
1282               -- Added MD4 code submited by Dobes Vandermeer (dobes@smartt.com)
1283               -- Added "burn_stack" function [idea taken from another source of c
1284                  alot of variables it will clean up better.  Functions like the e
1285                  stacks cleaned and the rest of the code is getting much more str
1286               -- Added a hashing demo by Daniel Richards (kyhwana@world-net.co.nz
1287               -- I (Tom) modified some of the test vector routines to use more ve
1288                  For example, the MD5/SHA1 code now uses all of the test vectors
1289               -- Fixed the register/unregister functions to properly report error
1290               -- Correctly updated yarrow code to remove a few unused variables.
```

```
1295        v0.46  -- Added in HMAC code from Dobes Vandermeer (dobes@smartt.com)
1296
1297        June 8th, 2002
1298        v0.45  -- Fixed bug in rc5.c where if you called rc5_setup() before regist
1299                  undefined behaviour.
1300               -- Fixed mycrypt_cfg.h to eliminate the 224 bit ECC key.
1301               -- made the "default" makefile target have depends on mycrypt.h and
1302
1303        Apr 4th, 2002
1304        v0.44  -- Fixed bug in ecc.c::new_point() where if the initial malloc fail
1305
1306        Mar 22nd, 2002
1307        v0.43  -- Changed the ZLIB code over to the 1.1.4 code base to avoid the "
1308               -- Updated the GCC makefile not to use -O3 or -funroll-loops
1309               -- Version tag in mycrypt.h has been updated :-)
1310
1311        Mar 10th, 2002
1312        v0.42  -- The RNG code can now use /dev/urandom before trying /dev/random
1313
1314        Mar 3rd, 2002
1315        v0.41  -- Added support to link and use ciphers at compile time.  This can
1316               -- Added a demo to show off how small an application can get... 46k
1317               -- Disastry pointed out that Blowfish is supposed to be high endian
1318               -- Made registry code for the PRNGs as well [now the smallest useab
1319
1320        Feb 11th, 2002
1321        v0.40  -- RSA signatures use [and check for] fixed padding scheme.
1322               -- I'm developing in Linux now :-)
1323               -- No more warnings from GCC 2.96
1324
1325        Feb 5th, 2002
1326        v0.39  -- Updated the XTEA code to work in accordance with the XTEA design
1327
1328        January 24th, 2002
1329        v0.38  -- CFB and OFB modes can now handle blocks of variable size like th
1330               -- Wrote a wrapper around the memory compress functions in Zlib tha
1331                  in the rest of my crypto lib
1332
1333        January 23rd, 2002
1334        v0.37  -- Added support code so that if a hash size and key size for a cip
1335                  use the next lower key supported.  (mainly for the PK code).  So
1336                  Twofish, etc...
1337               -- Added more options for Twofish.  You can now tell it to use prec
1338                  This will speed up the TWOFISH_SMALL implementation by increasin
1339               -- Fixed a bug in prime.c that would not use the correct table if y
1340               -- Fixed all of the PK packet code to use the same header format [s
1341                  binary wise incompatible with previous releases while the API ha
1342
1343        January 22nd, 2002
1344        v0.36  -- Corrections to the manual
1345               -- Made a modification to Twofish which lets you build a "small ram
1346                  about 190 bytes of ram for the key storage compared to the 4,200
1347                  variant requires.
1348               -- Reduced the stack space used in all of the PK routines.
1349
1350        January 19th, 2002
1351        v0.35  -- If you removed the first hash or cipher from the library it woul
1352                  you used an ID=0 [i.e blowfish or sha256] in any routine.  Now i
```

```
1357                Originally I wanted to ensure that the keys were smaller than th
1358                However, the bias is so insignifcant using full sizes.  For exam
1359                is about 2^191.99, so instead I rounded down and used a 184-bit
1360                key the code will work just the same except that some 192-bit ke
1361                deal since 1/2^192 is a very small bias!
1362             -- Made the configuration a bit simpler and more exacting.  You can
1363                key settings you wish to support without including the data for
1364                in a new file called "mycrypt_cfg.h"
1365             -- Configured "mpi-config.h" so its a bit more conservative with th
1366             -- Jason Klapste submitted bug fixes to the yarrow, hash and variou
1367                use what ever remaining hash/cipher combo is left [after you #un
1368                a fix to remove unused structures from the symmetric_key and has
1369             -- Made the CTR code handle variable length blocks better. It will
1370                encrypt messages any size block at a time.
1371             -- Simplified the yarrow code to take advantage of the new CTR code
1372             -- Added a 4096-bit DH key setting.  That took me about 36 hours to
1373             -- Changed the base64 routines to use a real base64 encoding scheme
1374             -- Added in DH and ECC "encrypt_key()" functions.  They are still r
1375             -- Added **Twofish** to the list of ciphers!
1376
1377       January 18th, 2002
1378       v0.34  -- Added "sha512" to the list of hashes.  Produces a 512-bit messag
1379                padding with the rsa_sign() function you cannot use sha512 with
1380             -- Cleaned up the other hash functions to use the LOAD and STORE ma
1381
1382       January 17th, 2002
1383       v0.33  -- Made the lower limit on keysizes for RSA 1024 bits again because
1384                work with the padding scheme and large symmetric keys.
1385             -- Added information concerning the Zlib license to the manual
1386             -- Added a 3072-bit key setting for the DH code.
1387             -- Made the "find_xyz()" routines take "const char *" as per Clay C
1388             -- Fixed an embarassing typo in the manual concerning the hashes.
1389             -- Fixed rand_prime() so that it makes primes bigger than the setti
1390                if you want a 1024-bit prime it would make a 1023-bit one.  Now
1391                it makes is always greater than 2^(8n) (n == bytes in prime).  T
1392                impact on security but I corrected it just the same.
1393             -- Fixed the CTR routine to work on platforms where char != 8-bits
1394             -- Fixed sha1/sha256/md5/blowfish to not assume "unsigned long == 3
1395                I "AND" with 0xFFFFFFFF.  That forces only the lower 32-bits to
1396                most compilers optimize out the AND operation since its a nop.
1397
1398       January 16th, 2002
1399       v0.32  -- Made Rijndael's setup function fully static so it is thread safe
1400             -- Svante Seleborg suggested a cosmetic style fixup for aes.c,
1401                basically to remove some of the #defines to clean it up
1402             -- Made the PK routines not export the ASCII version of the names o
1403                the PK message formats *incompatible* with previous releases.
1404             -- Merge in Zlib :-)
1405
1406
1407       January 15th, 2002
1408       v0.31  -- The RSA routines can now use CRT to speed up decryption/signatur
1409                compatible with previous releases.
1410             -- Fixed another bug that Svante Seleborg found.  Basically you cou
1411                rsa_exptmod() function itself if you're not careful.  That's fix
1412                rsa_exptmod() where if it knows the buffer you passed is too sma
1413                memory.
1414             -- improved the readability of the PK import/export functions
```

```
1419        v0.30  -- Major change to the Yarrow PRNG code, fixed a bug that Eugene St
1420                  Basically if you added entropy to the pool in small increments i
1421                  cancel out.  Now I hash the pool with the new data which is way
1422
1423        January 12th, 2002
1424        v0.29  -- Added MPI code written by Svante Seleborg to the library.  This
1425                  easier to follow and debug.  Actually I've already fixed a memor
1426               -- Memory leaks found and correct in all three PK routines.  The le
1427                  operation fails so it wouldn't normally turn up in the course of
1428               -- Fixed bugs in dh_key_size and ecc_key_size which would return ga
1429
1430        January 11th, 2002
1431        v0.28  -- Cleaned up some code so that it doesn't assume "char == 8bits".
1432                  changed.
1433               -- ***HUGE*** changes in the PK code.  I check all return values in
1434                  are errors [insufficient memory, etc..] it will be reported.  Th
1435                  robust and likely to catch any errors.
1436               -- Updated the is_prime() function to use a new prototype [it can r
1437                  does trial divisions against more primes before the Rabin Miller
1438               -- Added OFB, CFB and ECB generic wrappers for the symmetric cipher
1439               -- Added Xtea to the list of ciphers, to the best of my ability I h
1440                  I should note that there is not alot of concrete information abo
1441                  I found did not address endianess and were not even portable!.
1442                  best of my knowledge implements the Xtea algorithm as per the [s
1443               -- Reformated the manual to include the **FULL** source code optimi
1444
1445        January 9th, 2002
1446        v0.27  -- Changed the char constants to numerical values.  It is backwards
1447                  platforms where 'd' != 100 [for example].
1448               -- Made a change to rand_prime() which takes the input length as a
1449                  a negative len to get a "3 mod 4" style prime... oops
1450               -- changed the MSVC makefile to build with a warning level of three
1451
1452        January 8th, 2002
1453        v0.26  -- updated SHA-256 to use ROR() for a rotate so 64-bit machines won
1454                  the output
1455               -- Changed #include <> to #include "" for local .h files as per Ric
1456               -- Fixed bug in MPI [well bug in MSVC] that compiled code incorrect
1457                  I added a work around that catches the error and continues norma
1458
1459        January 8th, 2002
1460        v0.25  -- Added a stupid define so MSVC 6.00 can build the library.
1461               -- Big thanks to sci.crypt and "Ajay K. Agrawal" for helping me por
1462
1463        January 7th, 2002
1464        v0.24  -- Sped up Blowfish by unrolling and removing the swaps.
1465               -- Made the code comply with more traditional ANSI C standards
1466                  Should compile with MSVC with less errors
1467               -- moved the demos and documentation into their own directories
1468                  so you can easily build the library with other tool chains
1469                  by compiling the files in the root
1470               -- converted functions with length of outputs to use
1471                  "unsigned long" so 16-bit platforms will like this library more.
1472
1473        January 5th, 2002
1474        v0.23  -- Fixed a small error in the MPI config it should build fine anywh
1475
1476        January 4th, 2002
```

```
                  compressed I handled them incorrectly.

       January 4th, 2002
       v0.21  -- sped up the ECC code by removing redundant divisions in the
                 point add and double routines.  I also extract the bits more
                 efficiently in "ecc_mulmod()" now.
              -- sped up [and documented] the rand_prime() function.  Now it just
                 makes a random integer and increments by two until a prime is fo
                 This is faster since it doesn't require alot of calls to the PRN
                 it doesn't require loading huge integers over and over.  rand_pr
                 can also make primes congruent to 3 mod 4 [i.e for a blum intege
              -- added a gf_sqrt() function that finds square roots in a GF(2^w)
              -- fixed a bug in gf_div() that would return the wrong results if t
                 divisor than the dividend.

       January 4th, 2002
       v0.20  -- Added the fixed MPI back in so RSA and DH are much faster again

       v0.19  -- Updated the manual to reflect the fact that Brian Gladman wrote
              -- DH, ECC and RSA signature/decryption functions check if the key
              -- new DH signature/verification code works just like the RSA/ECC v

       January 3rd, 2002
       v0.18  -- Added way more comments to each .C file
              -- fixed a bug in cbc_decrypt(pt, ct, key) where pt == ct [i.e same
              -- fixed RC5 so it reads the default rounds out of the cipher_descr
              -- cleaned up ecc_export()
              -- Cleaned up dh_import() and ecc_import() which also perform more
                 error checking now
              -- Fixed a serious flaw in rsa_import() with private keys.

       January 2nd, 2002
       v0.17  -- Fixed a bug in the random prime generator that fixes the wrong b
              -- ECC and DH code verify that the moduli and orders are in fact pr
                 slows down the test routines alot but what are you gonna do?
              -- Fixed a huge bug in the mp_exptmod() function which incorrectly
                 values of p.  I replaced it with a slow function.  Once the auth
                 I will switch back.

       January 1st, 2002 [whoa new year!]
       v0.16  -- Improved GF division code that is faster.
              -- documented the GF code

       December 31st, 2001
       v0.15  -- A 1792-bit and 2048-bit DH setting was added.  Took me all night
                 find a 1792 and 2048-bit strong prime but what the heck
              -- Library now has polynomial-basis GF(2^w) routines I wrote myself
                 ECC over GF(2^w) later on....
              -- Fixed a bug with the defines that allows it to build in windows

       December 30th, 2001
       v0.14  -- Fixed the xxx_encrypt() packet routines to make an IV of appropr
                 for the cipher used.  It was defaulting to making a 256-bit IV..
              -- base64_encode() now appends a NULL byte, um "duh" stupid mistake
              -- spell checked the manual again... :-)

       December 30th, 2001
       v0.13  -- Switching back to older copy of MPI since it works! arrg..
```

```
1543                of the data to wipe.
1544
1545        December 29th, 2001
1546        v0.12  -- Imported a new version of MPI [the bignum library] that should
1547                  be a bit more stable [if you want to write your own bignum
1548                  routines with the library that is...]
1549              -- Manual has way more info
1550              -- hash_file() clears stack now [like it should]
1551              -- The artificial cap on the hash input size of 2^32 bits has been
1552                 removed.  Basically I was too lazy todo 64-bit math before
1553                 [don't ask why... I can't remember].  Anyways the hashes
1554                 support the size of 2^64 bits [if you ever use that many bits in
1555                 that's just wierd...]
1556              -- The hashes now wipe the "hash_state" after the digest is compute
1557                 prevent the internal state of the hash being leaked accidently [
1558
1559        December 29th, 2001
1560        v0.11  -- Made #define's so you can trim the library down by removing
1561                  ciphers, hashs, modes of operation, prngs, and even PK algorithm
1562                  For example, the library with rijndael+ctr+sha1+ECC is 91KB comp
1563                  to the 246kb the full library takes.
1564              -- Added ECC packet routines for encrypt/decrypt/sign/verify much a
1565                 the RSA packet routines.
1566              -- ECC now compresses the public key, a ECC-192 public key takes 33
1567                 for example....
1568
1569        December 28th, 2001
1570        v0.10  -- going to restart the manual from scratch to make it more
1571                  clear and professional
1572              -- Added ECC over Z/pZ.  Basically provides as much as DH
1573                 except its faster since the numbers are smaller.  For example,
1574                 A comparable 256-bit ECC key provides as much security as expect
1575                 from a DH key over 1024-bits.
1576              -- Cleaned up the DH code to not export the symbol "sets[]"
1577              -- Fixed a bug in the DH code that would not make the correct size
1578                 random string if you made the key short.  For instance if you wa
1579                 a 512-bit DH key it would make a 768-bit one but only make up 51
1580                 for the exponent... now it makes the full 768 bits [or whatever
1581                 is]
1582              -- Fixed another ***SERIOUS*** bug in the DH code that would defaul
1583                 keys by mistake.
1584
1585        December 25th, 2001
1586        v0.09  -- Includes a demo program called file_crypt which shows off
1587                  how to use the library to make a command line tool which
1588                  allows the user to encode/decode a file with any
1589                  hash (on the passphrase) and cipher in CTR mode.
1590              -- Switched everything to use typedef's now to clear up the code.
1591              -- Added AES (128/192 and 256 bit key modes)
1592
1593        December 24th, 2001
1594        v0.08  -- fixed a typo in the manual. MPI stores its bignums in
1595                  BIG endian not little.
1596              -- Started adding a RNG to the library.  Right now it tries
1597                 to open /dev/random and if that fails it uses either the
1598                 MS CSP or the clock drift RNG.  It also allows callbacks
1599                 since the drift RNG is slow (about 3.5 bytes/sec)
1600              -- the RNG can also automatically setup a PRNG as well now
```

```
1605                -- Fixed rsa_import to detect when the input
1606                   could be corrupt.
1607                -- added more to the manual.
1608
1609        December 22nd, 2001
1610        v0.06   -- Fixed some formatting errors in
1611                   the hash functions [just source code cleaning]
1612                -- Fixed a typo in the error message for sha256 :-)
1613                -- Fixed an error in base64_encode() that
1614                   would fail to catch all buffer overruns
1615                -- Test program times the RSA and symmetric cipher
1616                   routines for kicks...
1617                -- Added the "const" modifier to alot of routines to
1618                   clear up the purpose of each function.
1619                -- Changed the name of the library to "TomCrypt"
1620                   following a suggestion from a sci.crypt reader....
1621
1622        v0.05   -- Fixed the ROL/ROR macro to be safe on platforms
1623                   where unsigned long is not 32-bits
1624                -- I have added a bit more to the documentation
1625                   manual "crypt.pdf" provided.
1626                -- I have added a makefile for LCC-Win32.  It should be
1627                   easy to port to other LCC platforms by changing a few lines.
1628                -- Ran a spell checker over the manual.
1629                -- Changed the header and library from "crypt" to "mycrypt" to not
1630                   clash with the *nix package "crypt".
1631
1632        v0.04   -- Fixed a bug in the RC5,RC6,Blowfish key schedules
1633                   where if the key was not a multiple of 4 bytes it would
1634                   not get loaded correctly.
1635
1636        December 21st, 2001
1637
1638        v0.03   -- Added Serpent to the list of ciphers.
1639
1640        v0.02   -- Changed RC5 to only allow 12 to 24 rounds
1641                -- Added more to the manual.
1642
1643        v0.01   -- We will call this the first version.
1644
1645        /* $Source: /cvs/libtom/libtomcrypt/changes,v $ */
1646        /* $Revision: 1.288 $ */
1647        /* $Date: 2007/05/12 14:37:41 $ */
```