

Cloakware Corporation

**Presentation to the
Copy Protection Technical
Working Group**

Alec Main – VP Products and Services
alec.main@cloakware.com

September 11th 2002

Cloakware Corporation

- Based in Ottawa
- Founded in 1997
- 19 patents pending
- 26 personnel - 21 technical staff

www.cloakware.com

Additional References

- Cloakware whitepapers

<http://www.cloakware.com/resources/>

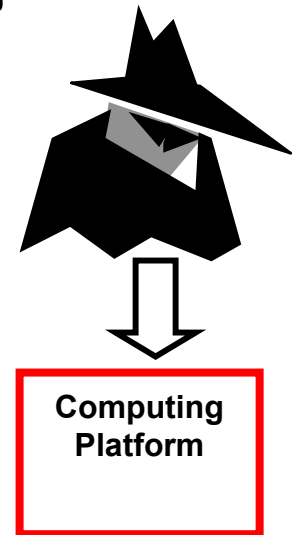
- White-box Cryptography and an AES Implementation, by S.Chow, P. Eisen, H. Johnson, P.C. van Oorschot, presented August 15-16, 2002 at the workshop on Selected Areas in Cryptography (SAC 2002). Includes a technical overview of white-box cryptography, the white-box attack context (WBAC), and an AES implementation of white-box cryptography.

<http://www.cloakware.com/pdfs/SAC2002-CW.pdf>

Code Transformation Technology

People Attacking Software

- Hacker has direct access to the software
- Abundant time and tools to conduct an attack
 - Decompilers, debuggers, emulators, hacker tools, dynamic analysis tools
- Highly skilled attacker
 - Reverse-engineer the code
 - Understand the code to extract secrets or algorithms
 - Find vulnerabilities (holes or bugs)
 - Tamper with the code (change its functionality)



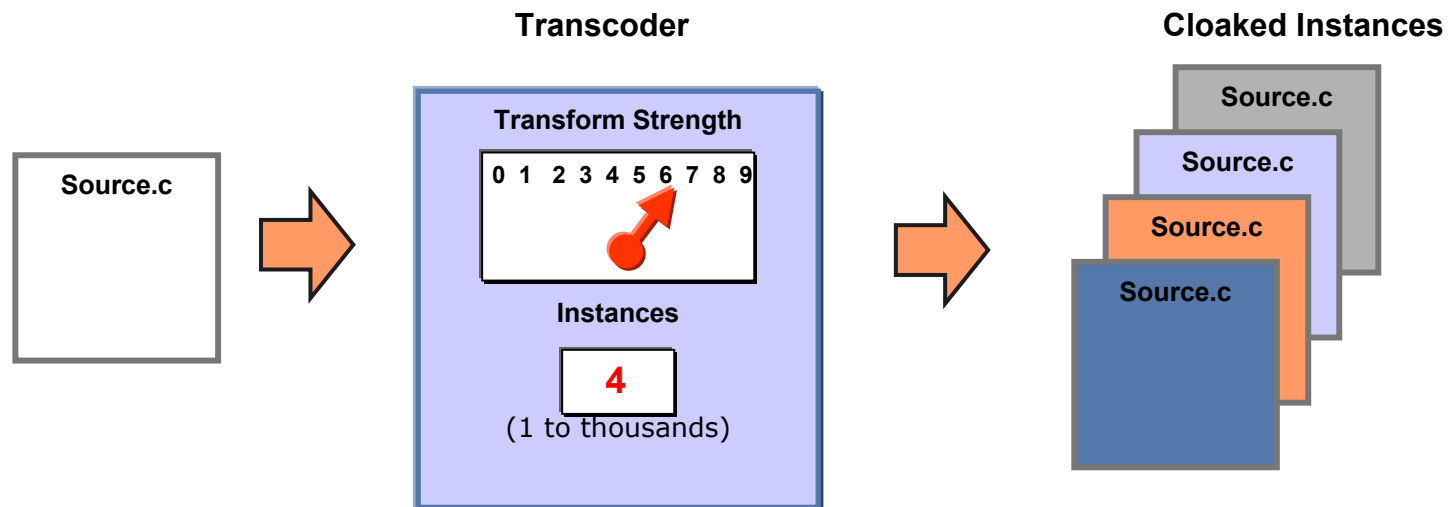
Software Attacking Software

- Automated attacks
 - Network penetration scripts/tools
 - Copy protection and decryption tools
- Easily propagated via the Internet
 - Posted on websites, BBS
 - Sent via e-mail, IRC
 - Viruses, Worms use a vector to propagate automatically
- Low skill level to use
 - Script kiddies
 - Point and click



Transcoder

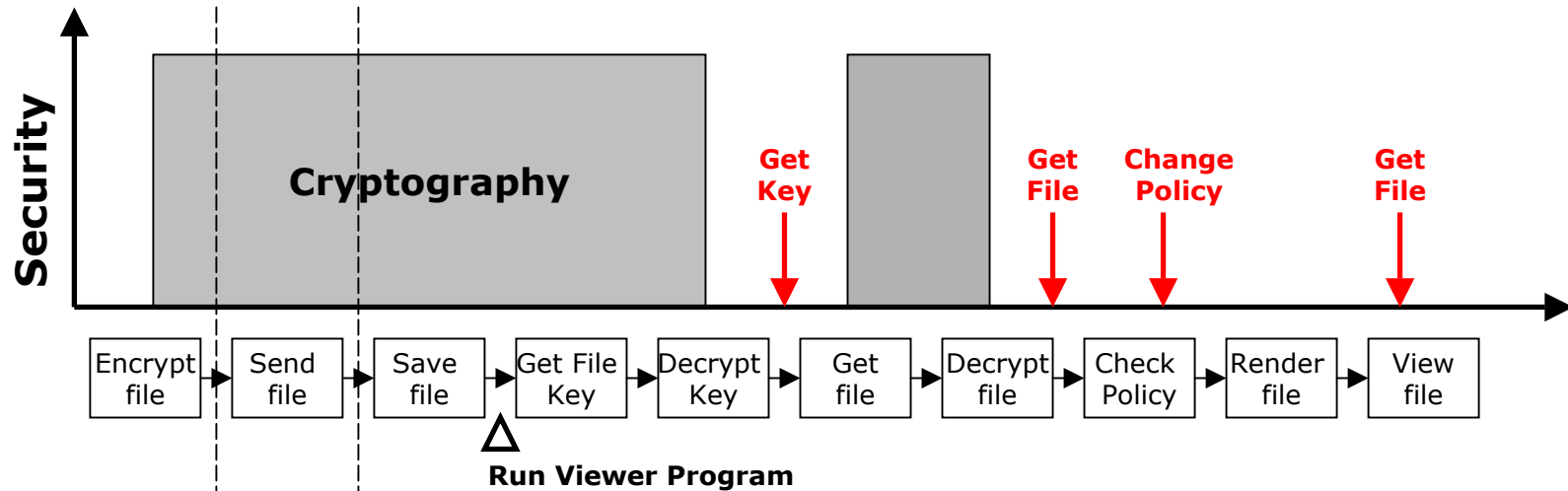
- Applies mathematical transformations to source code
- Fits into standard build processes and IDEs
- Transform strength determines resistance to tampering and reverse engineering attacks on a single program instance
- Diversity (# of instances) provides resistance to automated attacks



Code Transformation Features

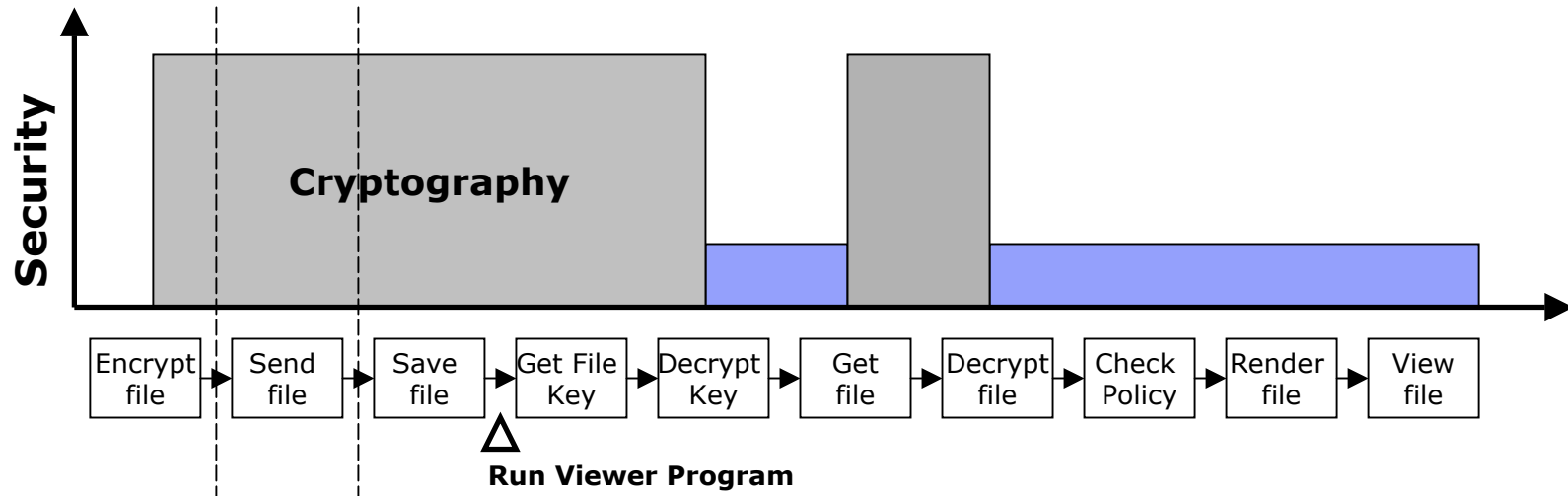
- Tamper Resistance (TR)
 - High security transforms create tamper-resistance
 - High security transforms increase code size and impact performance
 - Raises the skill required for reverse-engineering attacks
- Diversity
 - Changing random seed creates diverse copy
 - Reduces effectiveness of automated attacks
 - Exploit may exist, but attack prevented
 - Automated generation

Application Security with Crypto only



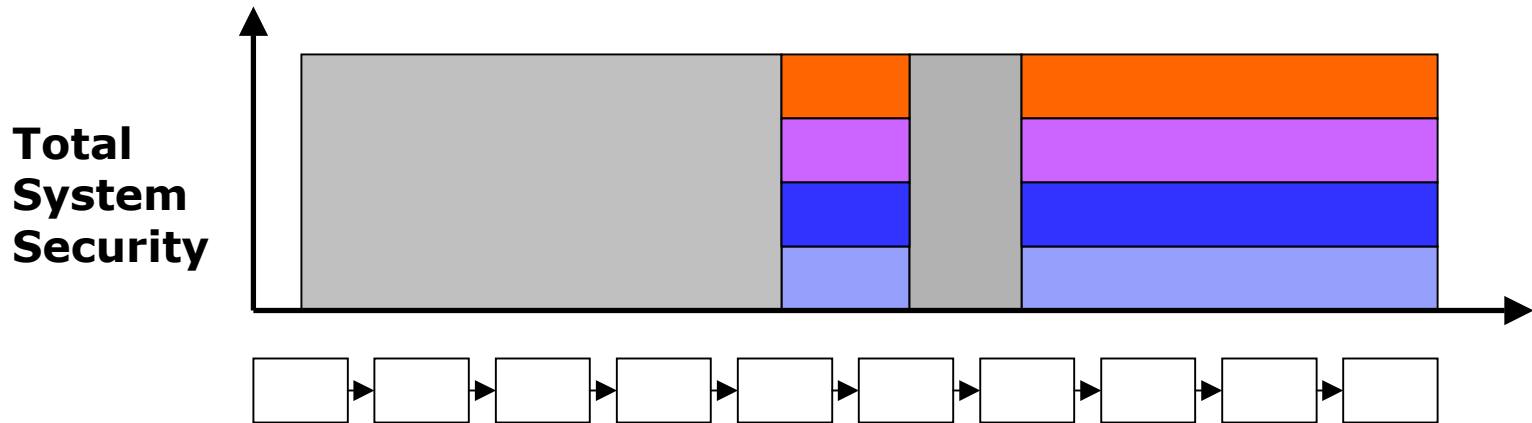
- Encryption only protects the data in storage and transit
- One of the fundamental flaws in application security

Application Security with Tamper Resistance



- Tamper Resistance raises the bar against direct access attacks
 - Increased time and skill required
 - Intrinsic to the code – ties data to code

Application Security with TR & Diversity



- Diversity raises the overall security
- Diversity prevents automated attacks

Code Transformations are Unbounded

- Under-lying mathematical basis provide an enormous range of techniques
 - Long evolutionary path
- Intrinsic to the application
 - No hardware or additional software required
- No portability constraints
- Flexible with fine grain control
- Complements good software engineering practices
 - Only object code is spaghetti
 - Automated

Security Metrics?

- An open technology
 - Academic and industry acceptance begun
- Papers on the theoretical upper bounds to reverse engineer transformed code
- White-Box Cryptography paper presented at Selected Areas of Cryptography (SAC) in August 2002
- Threat model:
 - Attacker has patents, know some transforms

Dr. Paul Van Oorschot
Cloakware Chief Scientist

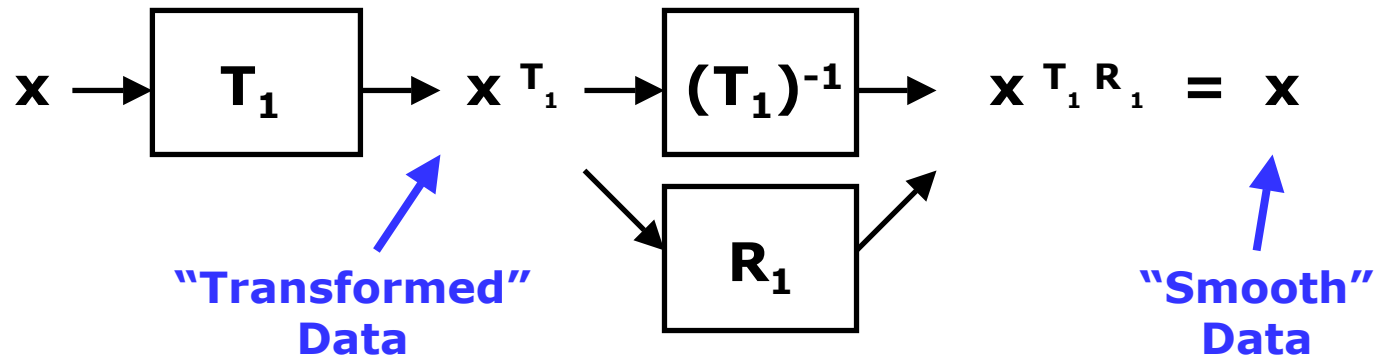
Former Entrust founding employee and Chief Scientist.
Co-author: *Handbook of Applied Cryptography*

What are Code Transformations?

- Data flow transforms
 - Transform data and operations
 - Increases time and skill required for reverse engineering
- Control flow transforms
 - Adds code with different properties for `Then` and `Else`
 - Increase resistance to tampering attacks
- White-box cryptography
 - Prevents key extraction from a crypto algorithm

Transforms

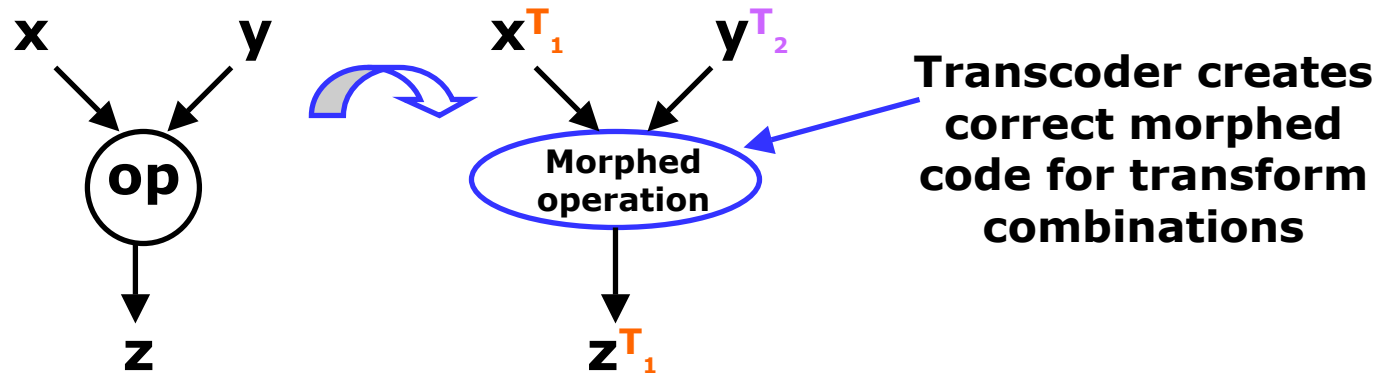
- Principle of data flow protection



- Many transforms possible
 - Linear, Quadratic, Finite Ring, Arbitrary Bitmap
 - Variable dependent









Transform Automatically Picked

- Transcoder applies transforms



- Automatically picked or specified by designer

Simple Linear Example

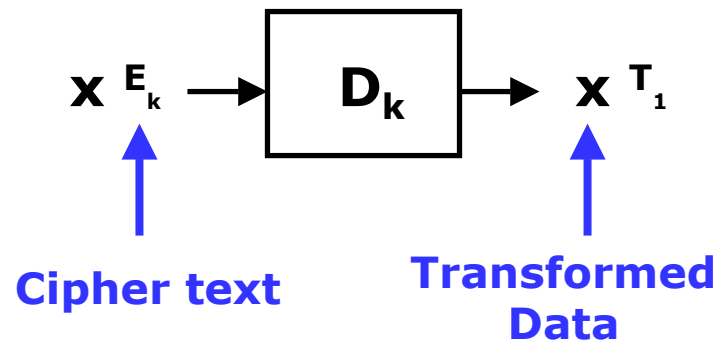
	Sample 1	Sample 2	Sample 3	Sample 4
Original code segment	$z = x + y$	$z = x + y$	$z = 2x + y$	$z = x + 5$
				
Transcoder transforms	$x' = 5x + 7$ $y' = 5y + 10$ $z' = 5z + 17$	$x' = 5x + 7$ $y' = -10y + 10$ $z' = 10z + 4$	$x' = 5x + 7$ $y' = -5y - 10$ $z' = 5z - 3$	$x' = 5x + 7$ $z' = 5z + 17$
				
Transformed code segment	$z' = x' + y'$	$z' = 2x' - y'$	$z' = 2x' - y'$	< no code >

Finite Ring

- Linear transforms, on a Finite Ring
 - i.e. Take advantage of overflow/wraparound
- Injects overflow into the transformed program
 - Original program must not overflow
 - Analysis tools and math packages choke
- Can divide by multiplying by inverse
- More diversity and security with less code expansion

White-Box Cryptography

- Cloakware provides AES library for development
 - Transcoder merges surrounding transforms and substitutes white box code
- Fixed key and dynamic key versions
 - ECB and CBC modes
- Transforms key and output and hides algorithm
- Crosses the transform boundary



The Transcoder

Language Support

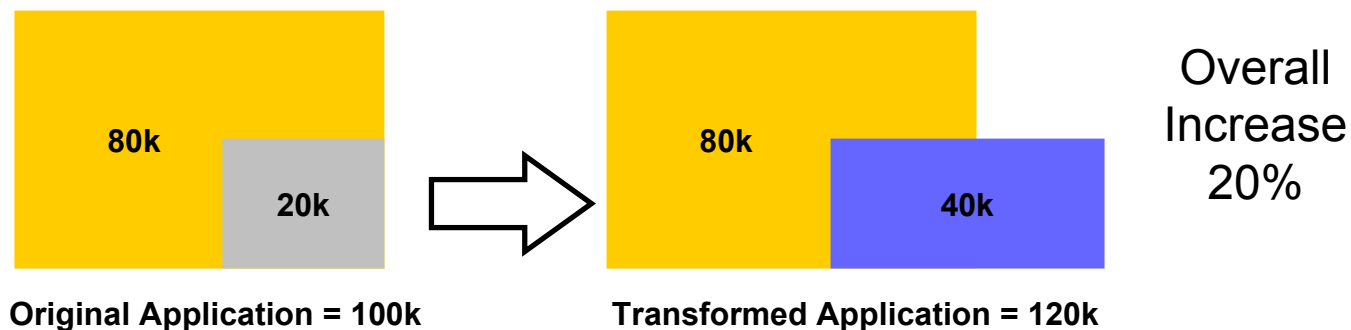
- C Language
 - Some language restrictions on transforms (e.g. no unions)
 - C++ coming
- Standard compilers supported
 - Microsoft Visual C on XP, 2000 and NT
 - GNU C for Intel on Linux Redhat 7.2 or higher
 - Metrowerks CodeWarrior PowerPC on Mac OS X
- Cloaked output typically not saved
 - Treated as an intermediate file in build process

Transcoder Control

- Config file provides default actions
- Source code annotations provide fine-grain control
 - e.g. `/** XC:Transform */` or `/** XC: Security Level(25) */`
 - Does not effect uncloaked builds
- Command line
 - Provides override options and supports build process
- IDE support
 - Add-in for MSVC 6.0
 - .NET and CodeWarrior support coming

Size and Performance Impact

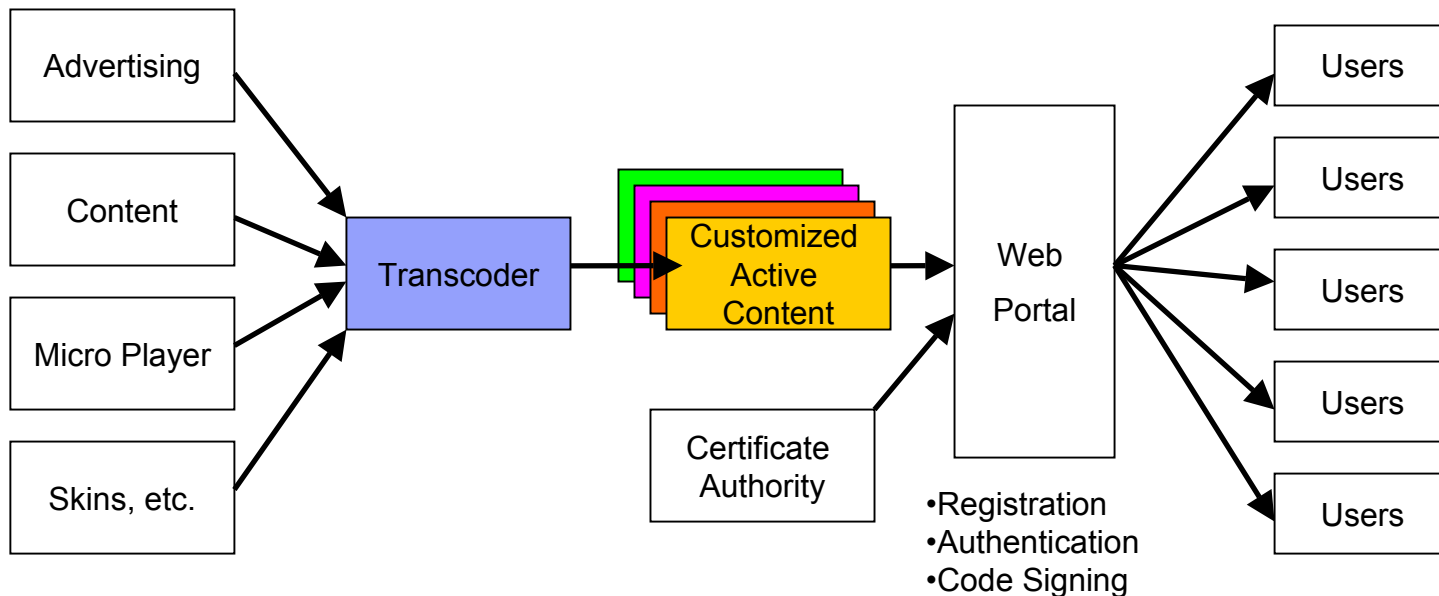
- Code expansion ranges from 10% to 400%
 - Only affects transformed files, typically security kernel
- Performance is highly dependant on source code
 - Controls exists to tune performance
 - Zero to 2x impact
 - White-box AES is fast, but greater code expansion



Applications Using Tamper Resistance and Diversity

Active Content

- Content, Advertising and Player all merged together
- Security not dependant on player install base



Customization

- Each customer gets a unique version
- Increases security delivered in single release cycle

