CryptoExperts

WE INNOVATE TO SECURE YOUR BUSINESS

White-box Cryptomania

Pascal Paillier

CryptoExperts

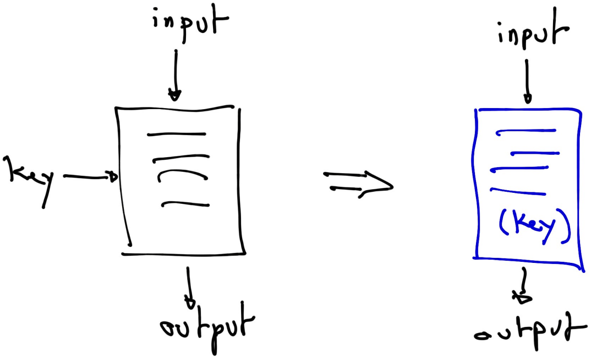
ECRYPT NET Workshop on Crypto for the Cloud & Implementation – Paris, June 27-28 2017

Overview

1. What is white-box crypto?
2. White-box compilers for signatures
3. White-box cryptomania
4. Conclusion: the lesson to learn
5. News from the front: the WhibOx Contest

What is white-box crypto?

The concept



CryptoExperts

What is NOT white-box crypto?

General purpose obfuscation

- from **any** program P, generate an obfuscated program $O(P)$
- hide **any** program property π in the code of $O(P)$
- meaning: the code of $O(P) \approx$ a black-box oracle that runs P

How realistic is obfuscation?

- **very** strong requirements on the compiler O
- known impossibility results (Barak et al, etc)

What is white-box crypto?

$\neq$ general program obfuscation!

White-box cryptography

- considers programs in a **restricted** class

$$\texttt{programs}(f) \quad \text{where} \quad f = \text{some keyed function}$$

- hides **some** program properties π in the code (but not all)
- code $\approx$ a black-box oracle **only in some adversarial contexts**
- already provably secure constructions for some f
- no impossibility results so far for f = blockcipher
- but **no secure** construction for e.g. $f = AES_k(\cdot)$, $k \leftarrow \$$

Overview

1. What is white-box crypto?
2. White-box compilers for signatures
3. White-box cryptomania
4. Conclusion: the lesson to learn
5. News from the front: the WhibOx Contest

White-box compilers for signatures

Let $\Sigma = (\mathit{KeyGen}, \mathit{Sign}, \mathit{Verif})$ be a public-key signature scheme.

Definition

A white-box compiler $\mathbf{C}_\Sigma$ takes a key pair $(sk, pk) \in \mathit{KeyGen}$ and some index $r \in \mathrm{R}$ and outputs a program $\mathbf{C}_\Sigma(sk, pk, r) = [\mathit{Sign}^r_{sk}]$.

Huge behavioral differences between

function $\mathit{Sign}(\cdot, \cdot)$	**oracle** $\mathit{Sign}(sk, \cdot)$	**program** $[\mathit{Sign}^r_{sk}]$
analytic description or algorithmic description	remote access, input/output only, typically stateful, private randomness	word in a language, stateless since rebootable, copiable, transferable, observable, modifiable, system calls simulatable
(specification)	(smart card)	(executable software)

A basic scheme: Schnorr signatures

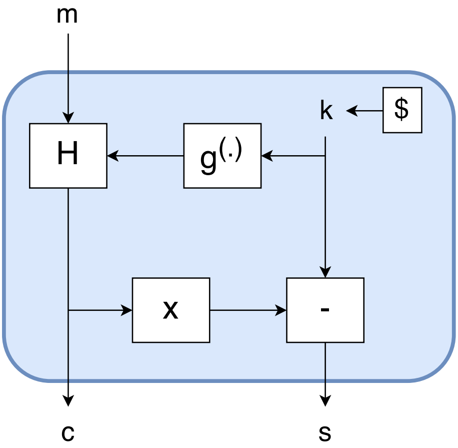
Pick some $\mathbb{G} = \langle g \rangle$ of order q.

$\mathit{KeyGen}(1^\kappa)$	$\mathit{Sign}(sk, m)$	$\mathit{Verif}(pk, m, (s, c))$
$x \leftarrow \mathbb{Z}_q$	$k \leftarrow \mathbb{Z}_q$	$H(m, g^s y^c) = c?$
$y = g^x$	$c = H(m, g^k)$	
	$s = k - cx \bmod q$	

- Existentially unforgeable in the ROM under the DL problem
- Known impossibility results in the SM

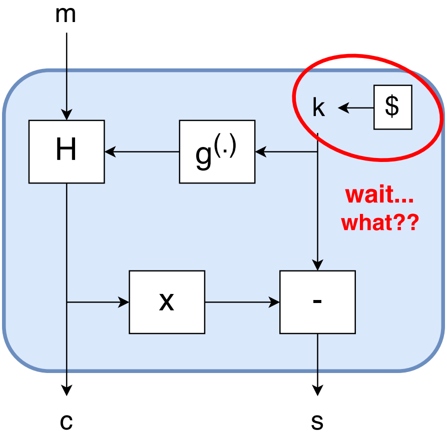
Schnorr signing programs

$[Sign^r_{sk}] =$



Schnorr signing programs

$[Sign^{r}_{sk}] =$



Schnorr signing programs

We intercept the call to the random source and put what we want

Then given the output (s, c)

$$x = \frac{k - s}{c}$$

This is a trivial break.

Schnorr signatures are not securely implementable as such

$k = \mathrm{PRNG}(m)$ not good enough either

$k = \mathrm{PRNG}(m, x)$ seems ok.

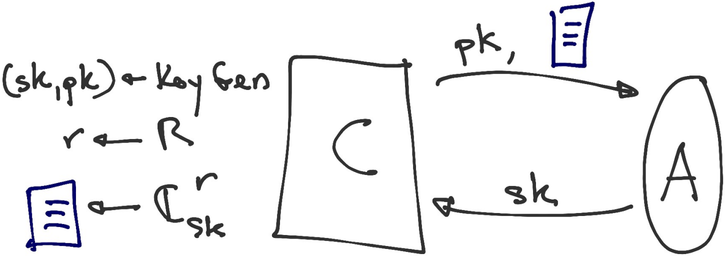
Overview

1. What is white-box crypto?
2. White-box compilers for signatures
3. **White-box cryptomania**
4. Conclusion: the lesson to learn
5. News from the front: the WhibOx Contest

White-box cryptomania

It's the world where $[Sign^r_{sk}]$ is safe and cozy.

What do we mean by that?



$\mathcal{A}$ does not exist unless inefficient.

Finally we have tamper-proof software for the Cloud!!

Security notions for signatures

$\alpha \Leftarrow \beta$: if β can be broken, α can be broken

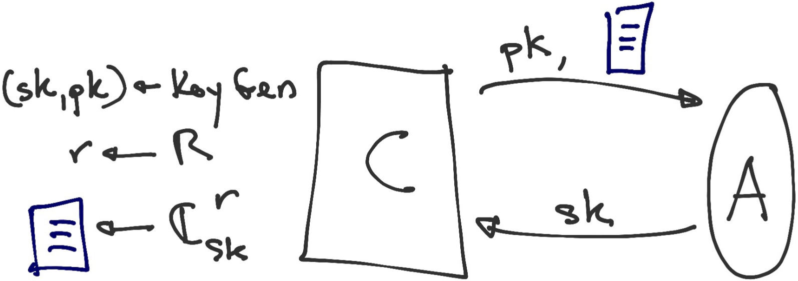
$$\begin{tikzcd}
\text{UBK-KOA} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \text{UUF-KOA} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \text{EUF-KOA} \arrow[d, Rightarrow] \\
\text{UBK-KMA} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \text{UUF-KMA} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \text{EUF-KMA} \arrow[d, Rightarrow] \\
\text{UBK-CMA} \arrow[r, Rightarrow] & \text{UUF-CMA} \arrow[r, Rightarrow] & \text{EUF-CMA}
\end{tikzcd}$$

But that's not sufficient to capture attack on programs.

Let's introduce **known program attacks**

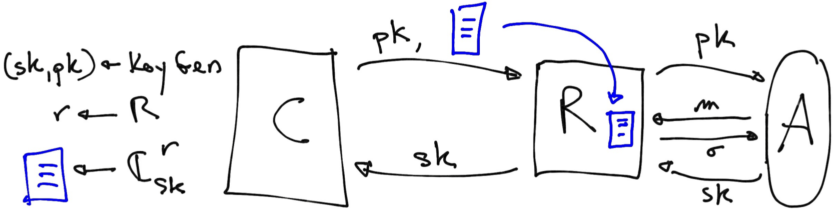
Known program attacks

UBK-KPA:



A first observation

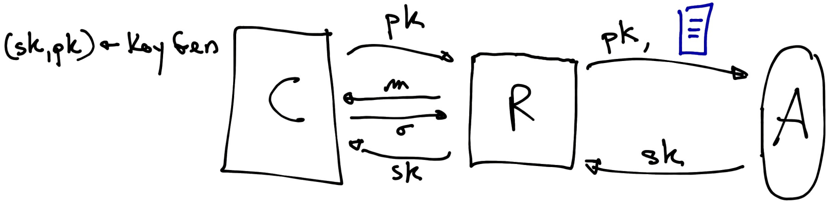
We have a reduction UBK-KPA $\Leftarrow$ UBK-CMA :



Equivalence CMA/KPA

In white-box cryptomania, we should loose nothing when switching from CMA to KPA.

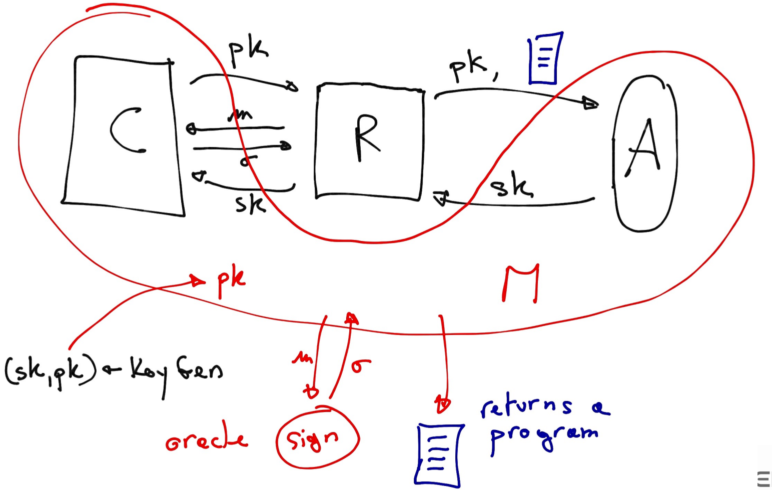
It means there must be a reduction in the other direction:



Now UBK-KPA = UBK-CMA :)

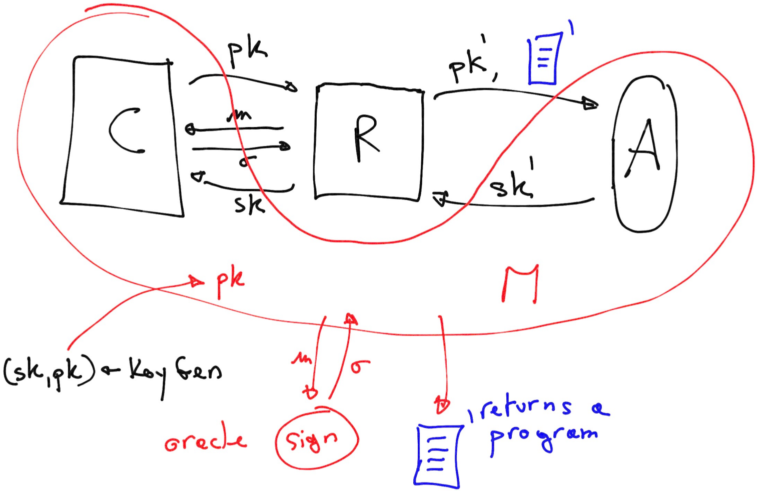
Program-reconstructing meta-reduction

We see that we can build a meta-reduction!



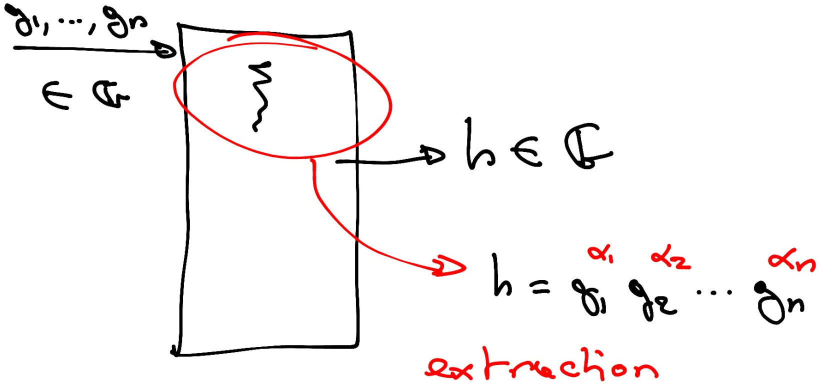
Program-reconstructing meta-reduction

… but the public-key given by $\mathcal{R}$ might be different from *pk*



Algebraic programs

“Algebraicity” over $\mathbb{G}$:



Huge class of algorithms, extends generic model

Repairing the biased program

If $\mathcal{R}$ is algebraic then we can extract the coefficients in

$$pk' = y' = g^{\alpha} y^{\beta}$$

so that given a program output (s', c') on m, we have

$$c' = H\left(m, g^{s'} y'^{c'}\right) = H\left(m, g^{s'} g^{\alpha c'} y^{\beta c'}\right)$$

If we

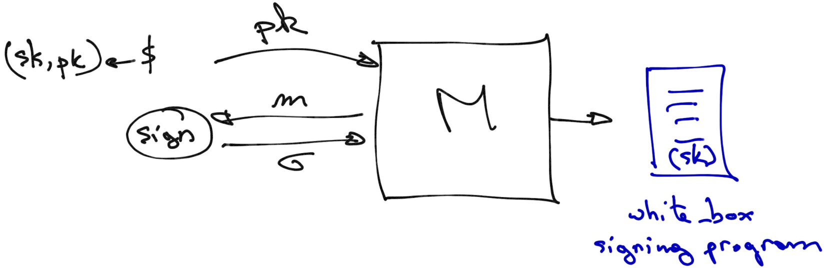
- pose $s = \frac{s' + \alpha c'}{\beta}$ and $c = c'$ and
- assume that generator g can be put into the public key pk,

then the program can be "repaired" into a signing program wrt the key pair (sk, pk) since

$$c = H\left(m, \left(g^{\beta}\right)^{s} \left(y^{\beta}\right)^{c}\right) \qquad pk = (g, y) \simeq (g^{\beta}, y^{\beta})$$

The effect of white-box cryptomania

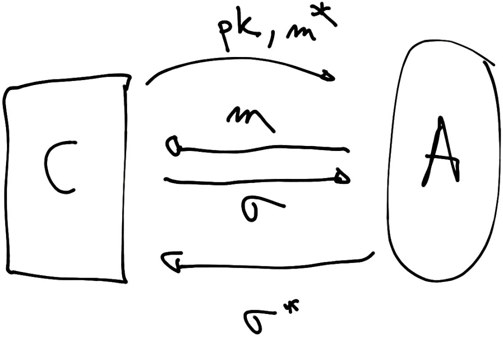
To summarize, white-box cryptomania gives us an efficient program reconstruction algorithm:



Impact on UUF-CMA

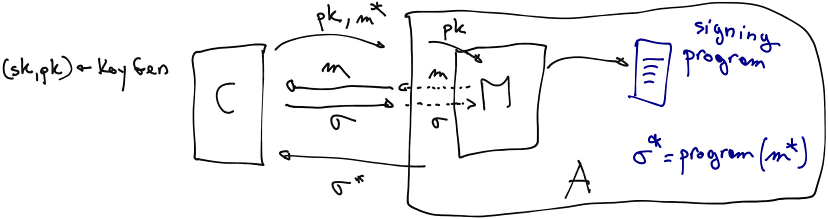
Recall the UUF-CMA game:

$(sk, pk) \leftarrow \text{KeyGen}$



Impact on UUF-CMA

Using $\mathcal{M}$, UUF-CMA is now easy to break :(



This is a huge collateral damage of white-box cryptomania, unavoidable unless we relax our definition of white-box cryptomania

Overview

1. What is white-box crypto?
2. White-box compilers for signatures
3. White-box cryptomania
4. **Conclusion: the lesson to learn**
5. News from the front: the WhibOx Contest

Conclusion: the lesson to learn

White-box crypto is a powerful paradigm

- beside the question of theoretic existence, the range of applications is immense
- white-box cryptomania is a bit too much: we do not want to loose the unforgeability properties of public-key signatures
- preferable to leave UBK-CMA and UBK-CPA non-equivalent to allow some security to subsist for UUF-CMA

This is work in progress

- a lot of questions remain
- can we have the same conclusions for e.g. ECDSA?
- how to relax white-box cryptomania?

Overview

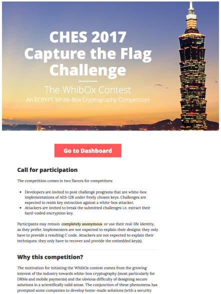
1. What is white-box crypto?
2. White-box compilers for signatures
3. White-box cryptomania
4. Conclusion: the lesson to learn
5. News from the front: the WhibOx Contest

News from the front: WhibOx Contest

CryptoExperts
WE INNOVATE TO SECURE YOUR BUSINESS

TU/e

ECRYPT CSA



CHES 2017
Capture the Flag
Challenge

The WhibOx Contest
An ECRYPT White-Box Cryptography Competition

Call for participation

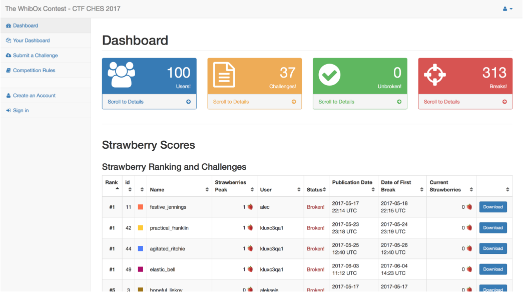
The competition comes in two flavors for competitors:

- Developers are invited to post challenge programs that are white-box implementations of AES-128 under freely chosen keys. Challenges are expected to resist key extraction against a white-box attacker.
- Attackers are invited to break the submitted challenges i.e. extract their hard-coded encryption key.

Participants may remain completely anonymous or use their real-life identity, as they prefer. Implementers are not expected to explain their designs: they only have to provide a resulting C code. Attackers are not expected to explain their techniques: they only have to recover and provide the embedded key(s).

Why this competition?

The motivation for initiating the WhibOx contest comes from the growing interest of the industry towards white-box cryptography (most particularly for DRMs and mobile payments) and the obvious difficulty of designing secure solutions in a scientifically valid sense. The conjunction of these phenomena has prompted some companies to develop home-made solutions (with a security



The WhibOx Contest - CTF CHES 2017

Dashboard

100 Users!

37 Challenges!

0 Unbroken!

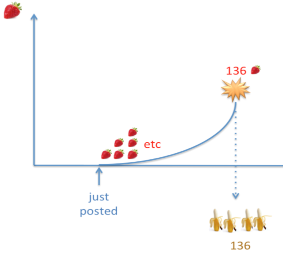
313 Breaks!

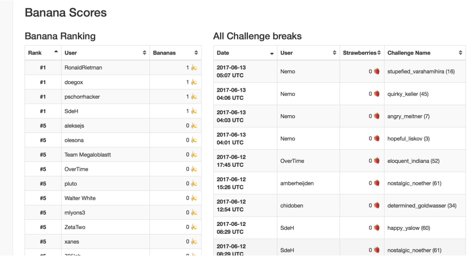
Strawberry Scores

Strawberry Ranking and Challenges

Rank	id		Name	Strawberries Peak	User	Status	Publication Date	Date of First Break	Current Strawberries	
#1	11	🟧	festive_jennings	1 🍓	alec	Broken!	2017-05-17 22:14 UTC	2017-05-18 22:15 UTC	0 🍓	Download
#1	42	🟨	practical_franklin	1 🍓	kluxc3qa1	Broken!	2017-05-23 23:18 UTC	2017-05-24 23:19 UTC	0 🍓	Download
#1	44	🟦	agitated_ritchie	1 🍓	kluxc3qa1	Broken!	2017-05-25 12:40 UTC	2017-05-26 12:40 UTC	0 🍓	Download
#1	49	🟪	elastic_bell	1 🍓	kluxc3qa1	Broken!	2017-06-03 11:12 UTC	2017-06-04 14:23 UTC	0 🍓	Download
#5	3	🟫	hopeful_liskov	0 🍓	alekseis	Broken!	2017-05-17	2017-05-17	0 🍓	Download

News from the front: WhibOx Contest





Banana Scores

Banana Ranking

Rank	User	Bananas
#1	RonaldRietman	1 🍌
#1	doegox	1 🍌
#1	pschorrhacker	1 🍌
#1	SdeH	1 🍌
#5	aleksejs	0 🍌
#5	olesona	0 🍌
#5	Team Megaloblastt	0 🍌
#5	OverTime	0 🍌
#5	pluto	0 🍌
#5	Walter White	0 🍌
#5	mlyons3	0 🍌
#5	ZetaTwo	0 🍌
#5	xanes	0 🍌
#5	[illegible]	0 🍌

All Challenge breaks

Date	User	Strawberries	Challenge Name
2017-06-13 05:07 UTC	Nemo	0 🍓	stupefied_varahamihira (16)
2017-06-13 04:06 UTC	Nemo	0 🍓	quirky_keller (45)
2017-06-13 04:03 UTC	Nemo	0 🍓	angry_meitner (7)
2017-06-13 04:01 UTC	Nemo	0 🍓	hopeful_liskov (3)
2017-06-12 17:45 UTC	OverTime	0 🍓	eloquent_indiana (52)
2017-06-12 15:26 UTC	amberheijden	0 🍓	nostalgic_noether (61)
2017-06-12 12:54 UTC	chidoben	0 🍓	determined_goldwasser (34)
2017-06-12 08:29 UTC	SdeH	0 🍓	happy_yalow (60)
2017-06-12 08:29 UTC	SdeH	0 🍓	nostalgic_noether (61)

https://whibox.cr.yp.to