



cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

New methods of prevention of cloning of tiny artefacts

Mirosław Kutylowski

Wrocław University of Technology
joint work with P. Błaśkiewicz and P. Kubiak

PERADA Workshop, Rome, 23.11.2010



cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Cryptography in the Real World



Security based on secrets

hardware units

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Reality

- 1 cryptography is a powerful tool, but...
- 2 it is based on secrecy of keys
- 3 machines participate in cryptographic protocols and not the humans (except for a password, PIN, ...)

so we have to trust black box devices for:

- safe storing the secret keys,
- performing the operations

Solutions

smart cards, TPMs, HSMs, SIM cards, Pay-TV decoders, ...



Black Side of Cryptography

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Cheating the user

- 1 side channel analysis, fault analysis, ... – extracting secrets from “tamper-proof” devices
- 2 installing trapdoors - from simple ones to advanced kleptographic techniques
- 3 logistic chain - how do you know that the chip you get comes from the claimed source?
- 4 how can you trust declaration of the manufacturer and/or certification body and/or authority?



Challenge

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

- How to demand from the citizens to trust the technology, when we know that it is not really secure?
- How to escape high costs or junky implementations?

For the rest of the talk we concentrate on *secure devices* creating electronic signatures.



cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Generating Cryptographic Keys



Generating cryptographic keys dilemma

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Generation directly on a signature creation device

- 1 if randomness not really random, then the keys might be really weak ...
- 2 ... but it is hardly possible to check that the randomness is really good
- 3 all kinds of kleptographic techniques apply

External generation

- 1 source of randomness could be of very good quality
- 2 easy to control and protect against installing trapdoors in the keys
- 3 ... **as long as trapdoors are not a feature of the system!**



Generating cryptographic keys dilemma

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Whom to trust?

- 1 manufacturer?
- 2 or service provider?

**what to do in case of a small country without control
over chip production and citizens not trusting blindly
own authorities?**



Nested signatures

solution idea

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Solution proposed

- standard RSA signatures, “backwards compatible”
- RSA keys generated externally



Nested signatures

solution idea

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Solution proposed

- standard RSA signatures, “backwards compatible”
- RSA keys generated externally
- ... but something hidden in the padding data
- namely: a deterministic signature (BLS) created with the keys generated by the signing device



Nested signatures

solution idea

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Solution proposed

- standard RSA signatures, “backwards compatible”
- RSA keys generated externally
- ... but something hidden in the padding data
- namely: a deterministic signature (BLS) created with the keys generated by the signing device

Main properties:

- now, in order to forge a signature collusion of the manufacturer and provider generating the keys is necessary:
 - the manufacturer might have access to the internal keys, but not to the RSA keys
 - the provider might retain the RSA keys but has no access to the internal keys



Why deterministic schemes?

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

- 1 no randomness means no easy room for hidden channel
- 2 even deterministic RSA admits some randomness in padding
it might be an unnecessary risk!
- 3 deterministic signatures with Discrete Logarithm is not the standard way of using DL problem but they are possible



cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Floating Keys



Floating exponents

detection of cloned signature creation devices

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Basic idea - mediated signatures

use mediated RSA (Boneh, Ding, Tsudik, Wang):

- the secret exponent split into two parts: $d = d_1 + d_2$,
- d_1 stored in the signature creation device, d_2 stored by a server controlling signing activities
 d_2 might be computed as $H(K, e)$ where K private key of the server and e a public key of the user.
- signing: $sign(M) = H(M)^{d_1} \cdot H(M)^{d_2}$
- the server can effectively block creating signatures by a device



Floating exponents

detection of cloned signature creation devices

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Floating exponents

the exponents d_1 and d_2 might float:

- there is a dynamic offset, say h , of the exponents:
 - the signature creation device holds $d_1 + h$
 - the server holds $d_2 - h$
- during each interaction a small number c is agreed between the signature device and the server, and the offset is updated $h := h + c$.

if two devices with the same key interact with the server, then the offset becomes de-synchronized: this leads to detection of clones!



cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Two-Head Dragon



Dragons

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Bad Dragon (European view)

- 1 a dragon has two heads
- 2 a knight can cut one dragon's head at a time
- 3 ... but at the meantime the second head of the dragon burns the knight



Dragons

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Bad Dragon (European view)

- 1 a dragon has two heads
- 2 a knight can cut one dragon's head at a time
- 3 ... but at the meantime the second head of the dragon burns the knight

Good Dragon (Chinese view)

- 1 a dragon with two heads is in a signing device
- 2 with each signature one head says a half of a magic formula
- 3 once the whole formula is said, the signatures disappear

If a card with the dragon has been cloned and used a few times, then with high probability the signatures disappear.



Basic trick

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Properties of Discrete Logarithm signature schemes

- ElGamal signature of M :

- $r = g^k$,

- $s = k^{-1}(\text{Hash}(M) - r \cdot x)$

for a key pair x -private key, $y = g^x$ -public key

- disclosing k reveals the secret key x :

$$x = r^{-1}(\text{Hash}(M) - s \cdot k)$$

Idea

Use this property for designing clone-evident devices.



Two-Head Dragon Signing Device

protocol 1

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Keys

Each signing device holds 4 keys:

- 1 two independent secret keys K_{D_0}, K_{D_1} for a deterministic signing scheme SD
- 2 two independent secret keys K_{P_0}, K_{P_1} for a deterministic signing scheme SP , say ElGamal



Two-Head Dragon

protocol 1, signing

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Signing M by a device Dev

- 1 a bit b is generated at random,
- 2 Dev reads the current value t from a counter t_{Dev} , and increments it,
- 3 Dev computes deterministic signatures $k_0 = SD_{K_{D_0}}(t, 0)$ and $k_1 = SD_{K_{D_1}}(t, 1)$,
- 4 if $b = 0$, then Dev computes a signature S_0 of M using $R(k_0)$ as the random parameter:
 $S_0 = SP_{K_{P_0}}(H(M) || k_1 || t; R(k_0))$
and outputs $(0, S_0, k_1, t)$.
If $b = 1$, then Dev computes $S_1 = C_{K_{P_1}}(H(M) || k_0 || t; R(k_1))$ using $R(k_1)$ as the random parameter, and outputs $(1, S_1, k_0, t)$.



Two-Head Dragon

protocol 1, verification

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

for $(0, S_0, k_1, t)$

- 1 check k_1 as a signature SD of t with key KD_1
- 2 check S_0 as a signature SP of M with key KP_0

for $(0, S_1, k_0, t)$

- 1 check k_0 as a signature SD of t with key KD_0
- 2 check S_1 as a signature SP of M with key KP_1



Two-Head Dragon

protocol 1, cloning evidence

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Revealing the secret key

Assumption: $(0, S_0, k_1, t)$ and $(0, S_1, k_0, t)$ are available, say $(0, S_0, k_1, t)$ created by honest device and $(0, S_1, k_0, t)$ by a forger

- 1 attack S_0 taking advantage of the fact that k_0 is used as randomness for creating S_0 : compute K_{P_0}
- 2 publish K_{P_0} .

After publishing

- the device is not tamper proof, or service provider has retained the keys uploaded into the device, or there is a trapdoor,...

... in any case the device **must not** be regarded as *secure signature creation device*



Two-Head Dragon

properties

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Discussion

- 1** In case of cloning:
 - the secret key of the probabilistic scheme gets revealed.
 - the secret key of the deterministic scheme is not revealed
- 2** **The second scheme**, which is slightly more technical guarantees that in case of forgery:

the keys for probabilistic scheme as well as keys for the deterministic scheme get revealed



Two-Head Dragon

summary

cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Main points

- 1 the user need not to be afraid that the service provider cheats – illegal copies of keys cannot be used
- 2 since forgery is self-evident: **no need for certification, no need for audit, ... automatic security evaluation**



Bibliography

- **“Digital Signatures for e-Government – a Long-Term Security Architecture”**, P. Błaśkiewicz, P. Kubiak, M. Kutylowski, LNICST, journal version *China Communications E-Forensic* 2010, Shanghai
- **“Two-Head Dragon. Clone-Fail Signature Creation Devices”**, P. Błaśkiewicz, P. Kubiak, M. Kutylowski, to appear LNCS, INTRUST 2010, Beijing

Acknowledgment

Thanks for support for Foundation for Polish Science, MISTRZ Programme, EU within the 7th Framework Programme, contract 215270 (FRONTS), and Polish Ministry of Science and Higher Education



cloning
prevention

Cryptography
and Reality

Generating
Keys

Floating Keys

Two-Head
Dragon

Thanks for your attention!

Contact data

- 1 `Miroslaw.Kutyloowski@pwr.wroc.pl`
- 2 `http://kutyloowski.im.pwr.wroc.pl`
- 3 `+48 71 3202109, fax: +48 71 320 2105`