Deterministic Digital Signatures in Bitcoin - 3ac

Nicolas T. Courtois

- **U**niversity **C**ollege **L**ondon, UK

UCL

New Paper:

Private Key Recovery Combination Attacks: On Extreme Fragility of Popular Bitcoin Key Management, Wallet and Cold Storage Solutions in Presence of Poor RNG Events

cf.

eprint.iacr.org/2014/848/

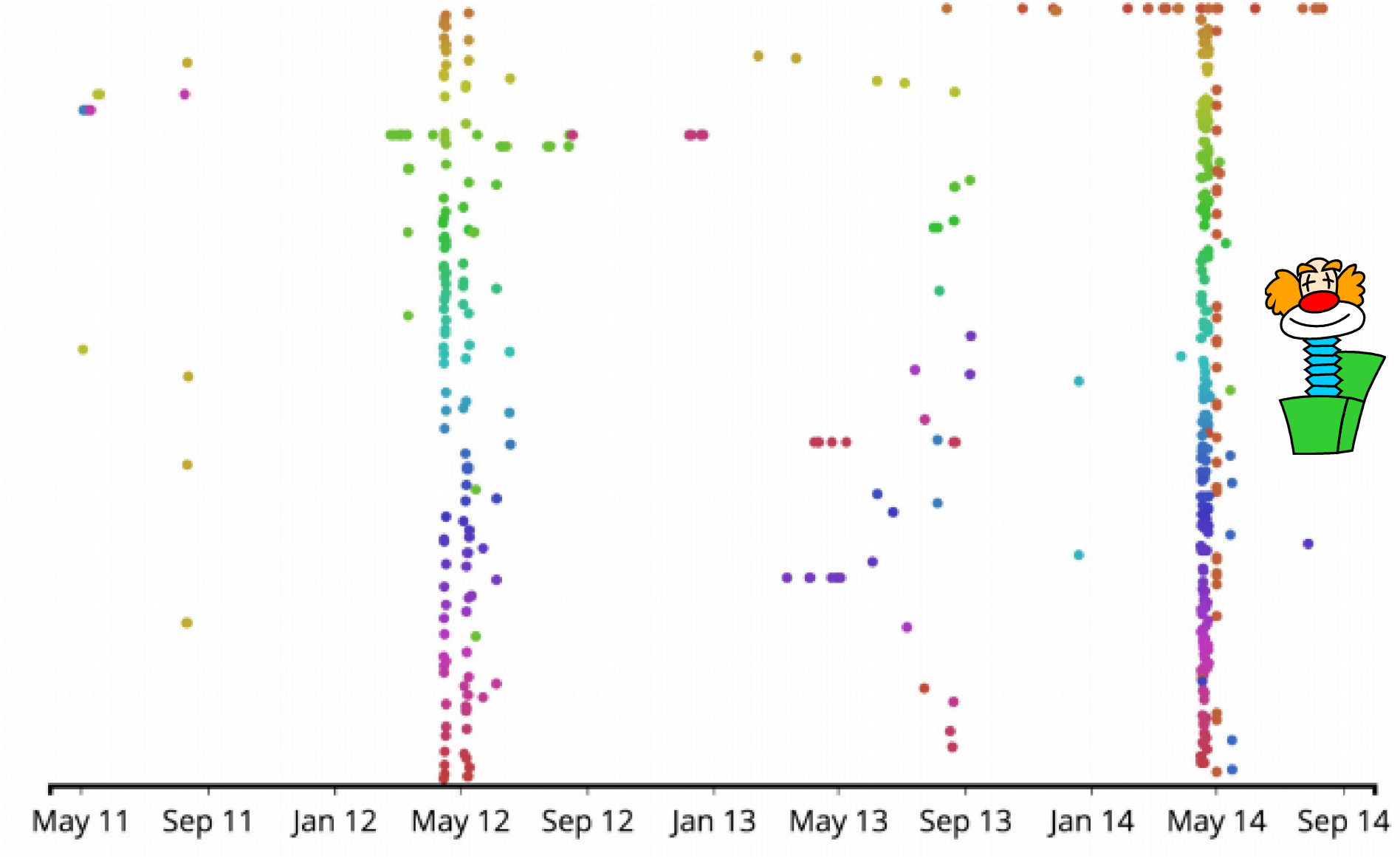
Nicolas T. Courtois[1] Pinar Emirdag[2] Filippo Valsorda[3]

[1] University College London, UK
[2] Independent market structure professional, London, UK
[3] CloudFlare, London, UK

Abstract. In this paper we study the question of key management and practical operational security in bitcoin digital currency storage systems. We study the security two most used bitcoin HD Wallet key management solutions (e.g. in BIP032 and in earlier systems). These systems have extensive audit capabilities but this property comes at a very high price. They are excessively fragile. One small security incident in a remote corner of the system and everything collapses, all private keys can be recovered and ALL bitcoins within the remit of the system can be stolen. Privilege escalation attacks on HD Wallet solutions are not new. In this paper we take it much further. We propose new more advanced **combination attacks** in which the security of keys hold in cold storage can be compromised without executing any software exploit on the cold system, but through security incidents at operation such as **bad random number or related random events**.

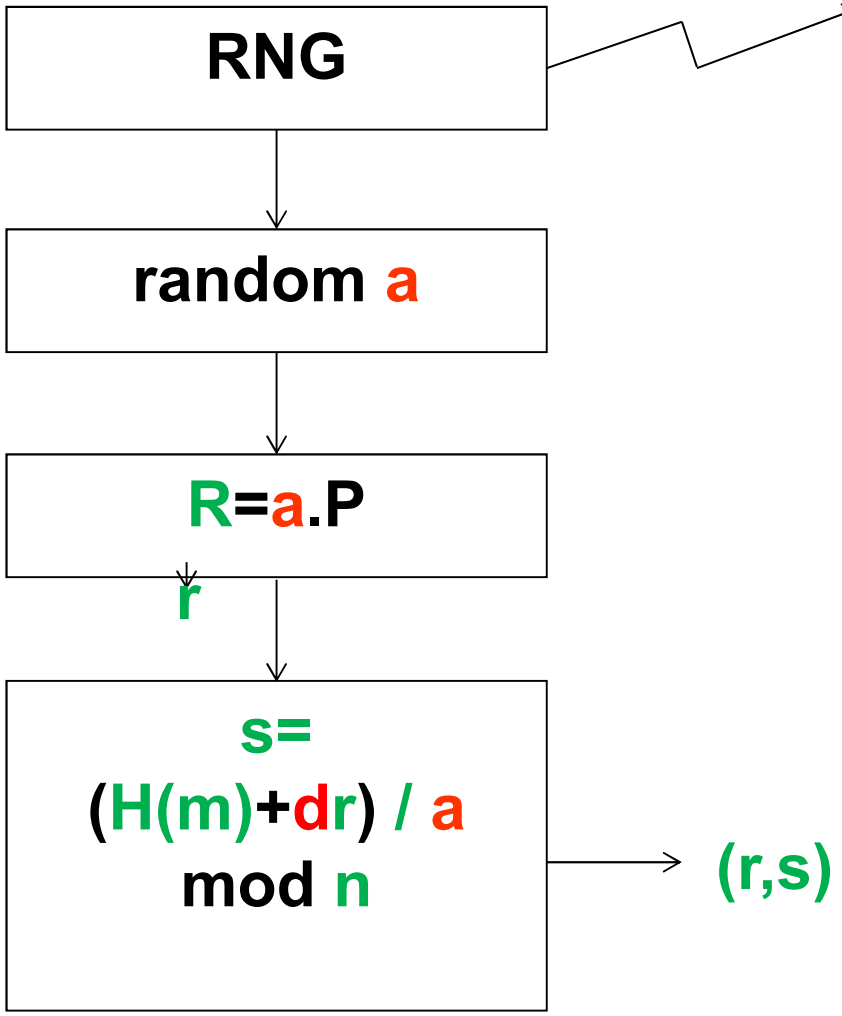
In our new attacks all bitcoins over whole large security domains can be stolen by people who have the auditor keys which are typically stored in hot systems connected to the Internet and can be stolen easily. Our combination attacks allow to recover private keys which none of the

Second Major Outbreak – May 2014



Attack – Same User

random a: must be kept secret!



has also happened 100 times in Bitcoin

same a used twice by the same user (d_1=d_2). In this case we have: (s_1a-H(m_1)) = rd =
(s_2a-H(m_2)) mod n
=> a=(H(m_1)- H(m_2))/(s_1-s_2) mod n AND now
d=(sa-H(m))/r mod n

anybody can steal the bitcoins!

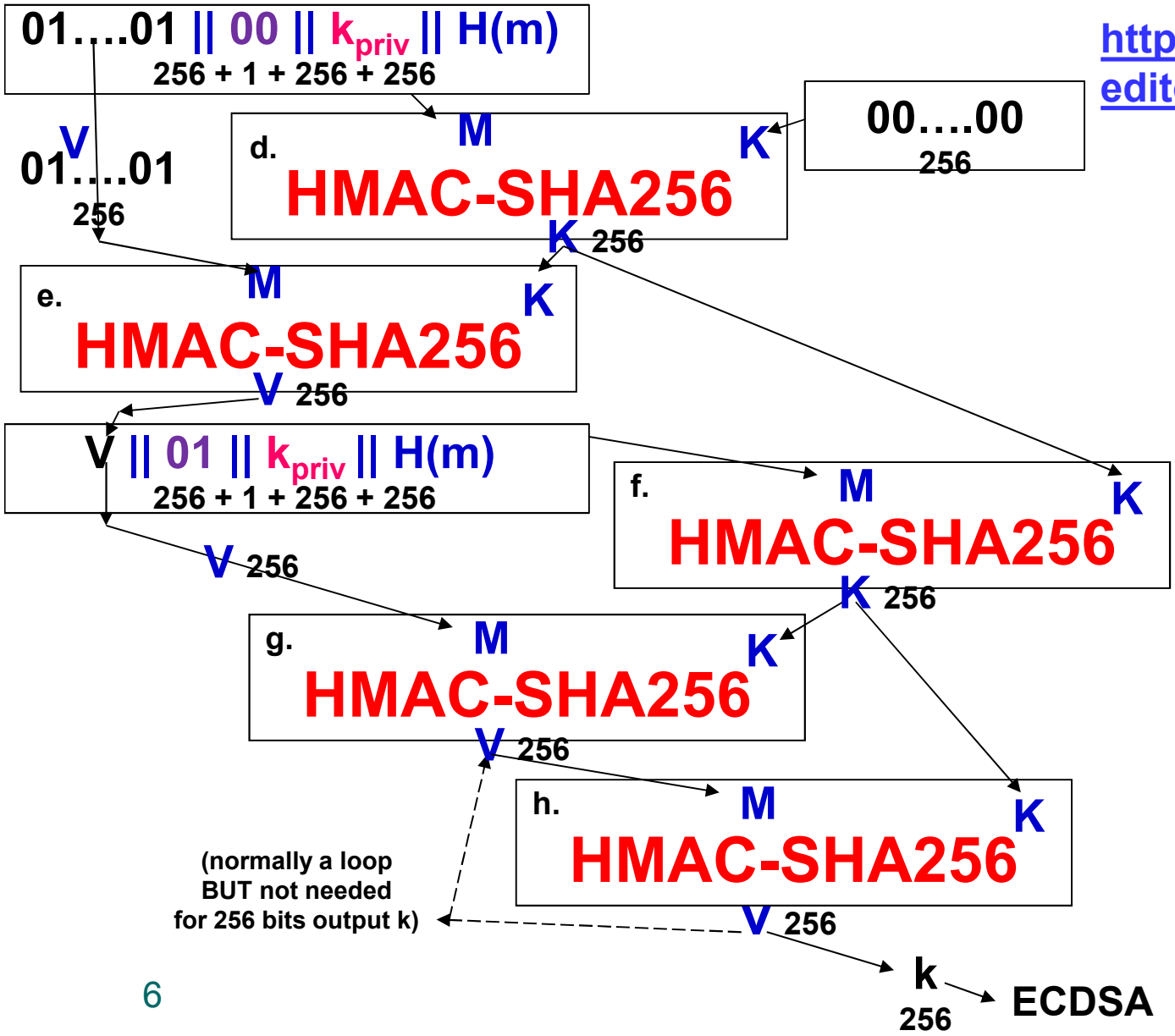
Is There a Fix?

Solution: RFC6979 [Thomas Pornin]

HOWEVER,

no existing cold storage solution
which have NOT already applied RFC6979
can claim to resist our attacks.

RFC6979 [Pornin] = 5+ applications of HMAC



http://www.rfc-editor.org/rfc/rfc6979.txt

HMAC-SHA256

Goal: avoids H(m||key) constructions
(which are subject to certain specific attacks)

Definition (from RFC 2104) [edit]

$$HMAC(K, m) = H\left((K \oplus opad) | H((K \oplus ipad) | m)\right)$$

where

H is a cryptographic hash function,

K is a secret key padded to the right with extra zeros to the input block size of the hash function, or the hash of the original key if it's longer than that block size,

512

m is the message to be authenticated,

| denotes concatenation,

⊕ denotes exclusive or (XOR),

opad is the outer padding (0x5c5c5c…5c5c, one-block-long hexadecimal constant),

and *ipad* is the inner padding (0x363636…3636, one-block-long hexadecimal constant).

HMAC-SHA256

Hashes twice with a key.

Definition (from RFC 2104)

$$HMAC(K, m) = H((K \oplus opad) | H$$

where

H is a cryptographic hash function,
K is a secret key padded to the right with the hash of the original key if it's longer th
m is the message to be authenticated,
| denotes concatenation,
⊕ denotes exclusive or (XOR),
opad is the outer padding (0x5c5c5c...5c
and *ipad* is the inner padding (0x363636.