Identity Card Key Generation in the Malicious Card Issuer Model

Arnis Paršovs

MTAT.07.022: Research Seminar in Cryptography

May 27, 2014

Problem Satement

- Private key generated by card issuer
- Cardholder liable for actions performed with private key
- Malicious card issuer can attack cardholder:
  - Copy private key before loading into smart card
  - Make random number generator predictable
  - Leak private key through side channel / backdoor
  - Leak private key through signature

Daniel J. Bernstein, Yun-An Chang, Chen-Mou Cheng, Li-Ping Chou, Nadia Heninger, Tanja Lange, and Nicko van Someren.
Factoring RSA Keys from Certified Smart Cards: Coppersmith in the Wild.
In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT (2)*, volume 8270 of *Lecture Notes in Computer Science*, pages 341–360. Springer, 2013.

Keys Generated by Cardholder

Solution: private key generation and storage sole responsibility of cardholder

- Cardholder must be protected against himself (against his compromised computer)

Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures. Official Journal L13/12, 1999.

Requirements for secure signature-creation devices
1. Secure signature-creation devices must, by appropriate technical and procedural means, ensure at the least that:
(a) the signature-creation-data used for signature generation **can practically occur only once**, and that their secrecy is reasonably assured;
(b) the signature-creation-data used for signature generation cannot, with reasonable assurance, be derived and the signature is protected against forgery using currently available technology;

(c) the signature-creation-data used for signature generation **can be reliably protected by the legitimate signatory against the use of others**.

Two Key Approach

- Key 1 – generated and stored on card issuer's smart card
- Key 2 – generated and stored on cardholder's device
- Document has to be signed by both keys to be valid

- Requires changes in protocols, standards and legislation

Can the same security be achieved using a single key?

Threshold RSA

- Trusted party generates RSA public key n, e and private key d
- RSA signing: $s = m^d$
- Trusted party splits private key d into two shares: $d = d_1 + d_2$
  - d_1 is loaded into smart card
  - d_2 is given to cardholder
- To produce signature:
  - $s_1 = m^{d_1}$ (calculated by smart card)
  - $s_2 = m^{d_2}$ (calculated by cardholder)
- $s = s_1 \cdot s_2 = m^{d_1} \cdot m^{d_2} = m^{d_1+d_2} = m^d$

Carmit Hazay, GertLæssøe Mikkelsen, Tal Rabin, and Tomas Toft.
Efficient RSA Key Generation and Threshold Paillier in the Two-Party Setting.
In *Topics in Cryptology – CT-RSA 2012*, volume 7178 of *Lecture Notes in Computer Science*, pages 313–331. Springer Berlin Heidelberg, 2012.

- 15 minutes for 2048-bit RSA on Intel Core i5 (semi-honest)

Abuse-free RSA Key Generation

- Key stored in card issuer's smart card
- Generated using a help from cardholder
- Neither card issuer nor cardholder learns the key
  - Two-party protocol
  - Cardholder's randomness included in the key
- Straightforward solution – threshold RSA
  - After generation cardholder's d_2 loaded into the smart card
  - Are there more efficient protocols?

Abuse-free RSA Key Generation: Verifiable Randomness

Yvo Desmedt.
Abuses in Cryptography and How to Fight Them.
In Shafi Goldwasser, editor, *Advances in Cryptology — CRYPTO' 88*, volume 403 of *Lecture Notes in Computer Science*, pages 375–389. Springer New York, 1990.

- Smart card has to prove that p and q are random primes
- Smart card chooses random string R_1 and discloses $E_{k_1}(R)$
- Cardholder chooses random string R_1' and sends to smart card
- Smartcard checks whether $R_1 \oplus R_1'$ is prime
  - If not – smart card opens commitment and protocol restarts
- Repeats protocol to generate q
- Finally the smart card proves using zero-knowledge: $PRIME(R_1 \oplus R_1') \wedge PRIME(R_2 \oplus R_2') \wedge n = (R_1 \oplus R_1')(R_2 \oplus R_2')$

Abuse-free RSA Key Generation: Verifiable Randomness

Ari Juels and Jorge Guajardo.
RSA Key Generation with Verifiable Randomness.
In David Naccache and Pascal Paillier, editors, *Public Key Cryptography*, volume 2274 of *Lecture Notes in Computer Science*, pages 357–374. Springer Berlin Heidelberg, 2002.

- Smart card and cardholder jointly select random integers x and y
- Using zero-knowledge range proof smart card proves that p and q lie in intervals $[x, x + l]$ and $[y, y + l]$
- Generating 2048-bit RSA would likely take over 40 minutes to execute on home router

Abuse-free RSA Key Generation: Multi-prime RSA

Generates 4096-bit 4-prime RSA key $n = p_1 q_1 p_2 q_2$

- p_1, q_1 generated and stored on smart card
- p_2, q_2 generated by cardholder and loaded into smart card

Security:

- For malicious card issuer and cardholder – 2048-bit RSA
- For any other attacker – 4096-bit RSA

Speed:

- Public key operations 4 times slower than 2048-bit RSA
- Private key operations 2 times slower if CRT used

Conclusion

Identity Card Key Generation in the Malicious* Card Issuer Model