RSAConference2018

San Francisco | April 16 – 20 | Moscone Center

#RSAC

MATTERS NOW

SESSION ID: Digital Signatures

REASSESSING SECURITY OF RANDOMIZABLE SIGNATURES

Olivier Sanders

Research Engineer
Orange Labs
Cesson-Sévigné, France

Joint work with **David Pointcheval**

erc

Agenda

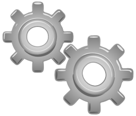
- Context
- Randomizable Signatures
- Our Contribution
- Conclusion

Context

Digital Signatures

Digital Signatures

electronic version of
handwritten signatures



building block

Expected Properties

Efficiency

Efficiency and Additional Features

Example: Anonymous Authentication

- Secure Devices (TPM, Intel SGX,...) may need to authenticate
- Standard Digital Signature is unsuitable:

  devices could be traced, which raises legal issues

- Such devices need (and use) anonymous authentication mechanisms

Zero-Knowledge Proofs

Anonymous authentication usually combines digital signatures with ZK proofs:

Primitive

Standard Model | Random Oracle Model

Groth-Sahai Proofs | Schnorr-like Proofs

better security | better efficiency

- Digital signature must interact smoothly with such proofs
- For practical uses, constructions in the ROM are unavoidable

Randomizability

- Complexity of ZK proofs increases with the number of elements to hide

  $\Rightarrow$ this number must be reduced as much as possible

- Randomizability allows to derive unlinkable versions σ' from a signature σ
- σ' can be shown

  $\Rightarrow$ significantly improves efficiency

Randomizable Signature

Camenisch-Lysyanskaya Signatures

- CL signatures achieve randomizability in a bilinear setting
  - $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ valid on $m \Rightarrow \forall t \in \mathbb{Z}_p, \sigma' = (\sigma_1^t, \sigma_2^t, \sigma_3^t)$ valid on m
  - σ and σ' are unlinkable under the DDH assumption

- Bilinear Groups: $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ of prime order p along with a map e such that
  - $\forall (g, \widetilde{g}) \in \mathbb{G}_1 \times \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p$ $e(g^a, \widetilde{g}^b) = e(g, \widetilde{g})^{a \cdot b}$
  - $e(g, \widetilde{g}) = 1_{\mathbb{G}_T} \implies g = 1_{\mathbb{G}_1}$ or $\widetilde{g} = 1_{\mathbb{G}_2}$

- Popular setting for privacy-preserving protocols:
  - Group Signature
  - Electronic Cash
  - ...

Example of Group Signature

- Join: Alice gets a signature $\sigma \leftarrow (\sigma_1, \sigma_2, \sigma_3)$ on her secret key sk $\in \mathbb{Z}_p$
- To anonymously prove membership in the group, Alice
  1. randomize σ: $t \xleftarrow{\$} \mathbb{Z}_p$, $\sigma' \leftarrow (\sigma_1^t, \sigma_2^t, \sigma_3^t)$
  2. sends σ' and proves that it is valid on the secret sk.
- Only sk needs to be hidden:

$\Rightarrow$ leads to very efficient protocols

Pointcheval-Sanders Signatures

- Complexity of CL-signatures increases with the number r of elements to sign
- PS signatures offer the same features with improved performances

	CL	PS
Size	$(1+2r)\ \mathbb{G}_1$	$2\ \mathbb{G}_1$
Sign Cost	$1\ \mathrm{R}_{\mathbb{G}_1} + 2r\ \mathbb{G}_1$	$1\ \mathrm{R}_{\mathbb{G}_1} + 1\ \mathbb{G}_1$
Verif Cost	$4r\ \mathrm{P} + r\ \mathbb{G}_2$	$2\ \mathrm{P} + r\ \mathbb{G}_2$
Randomizable	✓	✓

Security

- The standard security notion for signatures is EUF-CMA security:

$$
\begin{array}{lcl}
\textbf{Adversary} & & \textbf{Challenger } (\mathsf{sk}, \mathsf{pk}) \\
\hline
& \xleftarrow{\quad \mathsf{pk} \quad} & \\
& \xrightarrow{\quad m_1 \quad} & \sigma_1 \leftarrow \mathtt{Sign}(\mathsf{sk}, m_1) \\
& \xleftarrow{\quad \sigma_1 \quad} & \\
& \dots & \\
& \xrightarrow{\quad m_q \quad} & \sigma_q \leftarrow \mathtt{Sign}(\mathsf{sk}, m_q) \\
& \xleftarrow{\quad \sigma_q \quad} & \\
& \xrightarrow{\quad (m, \sigma) \quad} &
\end{array}
$$

- The adversary succeeds if σ is valid on m and $m \neq m_i$

Security

- The weaker EUF-wCMA security notion can be enough:

Adversary		Challenger
	$\xrightarrow{(m_1, \ldots, m_q)}$	$\sigma_i \leftarrow \mathtt{Sign}(\mathsf{sk}, m_i)$
	$\xleftarrow{(\mathsf{pk}, \sigma_1, \ldots, \sigma_q)}$	
	$\xrightarrow{(m, \sigma)}$	

- The adversary succeeds if σ is valid on m and $m \neq m_i$
- The messages are no longer adaptively chosen

Limits of Randomizable Signatures

- Randomizability of CL and PS signatures comes at a cost:

  security relies on the interactive LRSW and PS assumptions

- These assumptions essentially state the EUF-CMA security
- The lack of precise security assessment is an obstacle to a widespread deployment of these signatures
- Non-randomizable alternatives can be preferred for efficiency reasons

Boneh-Boyen Signature

- BB signatures is a popular (non-randomizable) alternative for privacy-preserving primitives
- EUF-CMA security relies on q-$\mathtt{SDH}$ assumption:

$$\text{given } (g, g^x, \ldots, g^{x^q}), \text{ it is hard to output } (w, g^{\frac{1}{x+w}}) \text{ with } w \in \mathbb{Z}_p^*$$

- q-$\mathtt{SDH}$ assumption is better accepted than interactive assumptions:
  - it is easier to assess
  - it is not directly related to EUF-CMA security

Our Contribution

q-MSDH Assumptions

$\mathbb{G}_1 = < g >$ and $\mathbb{G}_2 = < \tilde{g} >$

q-SDH

Asymmetric Pairings

$\boxed{q\text{-MSDH-1}}$

$$\{g^{x^i}, \tilde{g}^{x^i}\}_{i=0}^{q}, (g^a, \tilde{g}^a, \tilde{g}^{ax})$$

$$\downarrow$$

$$(w, P, h^{\frac{1}{x+w}}, h^{\frac{a}{P(x)}})$$

Symmetric Pairings

$\boxed{q\text{-MSDH-2}}$

$$\{g^{x^i}, g^{b \cdot x^i}\}_{i=0}^{q}, (g^a, g^{abx})$$

$$\downarrow$$

$$(w, P, h^{\frac{1}{x+w}}, h^{\frac{a}{xP(x)}})$$

return

- $h \in \mathbb{G}_1^*$, $w \in \mathbb{Z}_p$ such that $P(w) \neq 0$.
- We prove that they underlie EUF-wCMA security of PS and CL signatures

q-MSDH Assumptions

- We prove that both assumptions hold in the generic group model
- Intuition for q-MSDH-1:

$$\{g^{x^i}, \tilde{g}^{x^i}\}_{i=0}^{q}, (g^a, \tilde{g}^a, \tilde{g}^{ax}) \rightarrow (w, P, h^{\frac{1}{x+w}}, h^{\frac{a}{P(x)}})$$

  - randomizability implies that h can be any element of $\mathbb{G}_1^*$
  - $h^{\frac{a}{P(x)}}$ can be computed from g^a only if $h = g^{\lambda P(x)}$ with $\lambda \in \mathbb{Z}_p$
  - In such a case $h^{\frac{1}{x+w}} = g^{\frac{\lambda P(x)}{x+w}}$ cannot be computed from $\{g^{x^i}\}_{i=0}^{q}$ since $(x+w) \nmid P(x)$
- The same reasoning holds for q-MSDH-2:

EUF-wCMA Security

- These assumptions only underlie EUF-wCMA security of CL or PS signatures
- This security notion can be enough in some contexts if the certified secret value is generated collaboratively
- Example: in the Join procedure of a group signature scheme
  - Alice generates sk_1 and proves knowledge of it
  - the group manager selects and sends sk_2 along with a certificate on $sk_1 + sk_2$
  - Alice sets sk as $sk_1 + sk_2$

Achieving EUF-CMA Security

- CL and PS signatures can sign several messages

Signer(sk, pk)		User(pk)
	$\xleftarrow{(m_1, \ldots, m_r)}$	
$\sigma \leftarrow \texttt{Sign}(\mathsf{sk}, (m_1, \ldots, m_r))$	$\xrightarrow{(\sigma)}$	

Achieving EUF-CMA Security

- CL and PS signatures can sign several messages

Signer(sk, pk)		User(pk)
	$\xleftarrow{(m_1, \ldots, m_r)}$	
$m' \xleftarrow{\$} \mathbb{Z}_p$		
$\sigma \leftarrow \texttt{Sign}(\mathsf{sk}, (m_1, \ldots, m_r, m'))$	$\xrightarrow{(\sigma, m')}$	

- Signing an additional message m' is enough to achieve EUF-CMA security under these assumptions

- Slight increase of the complexity

Keeping Randomizability

- The additional message m' adds an element to the signature
- m' cannot be randomized
- In the ROM, we can set $m' = H(m_1, \ldots, m_r)$

  $\Rightarrow$ CL and PS features are kept

- No need to check that $m' = H(m_1, \ldots, m_r)$ in the verification process

  $\Rightarrow$ compatibility with ZK proofs is ensured

- Most protocols based on CL and PS signatures already use the ROM

Conclusion

Conclusion

- We reassessed security of CL and PS signatures and showed that:
  - they are EUF-wCMA secure under variants of q-$\mathtt{SDH}$ assumptions
  - they are EUF-CMA secure under the same assumptions assuming slight modifications
- We prove that these assumptions hold in the generic groups model
- CL or PS signatures can be used without jeopardizing security

  $\Rightarrow$ no need to choose between security and randomizability thank you

#RSAC

RSA Conference2018

San Francisco | April 16 – 20 | Moscone Center

MATTERS NOW

SESSION ID: CRYP-R12

DIFFERENTIAL ATTACKS ON DETERMINISTIC SIGNATURES

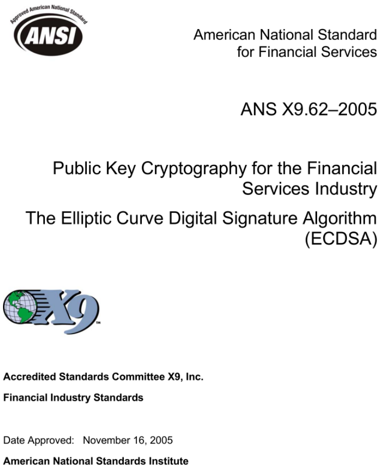
Marc Joye

Security Technologist
NXP Semiconductors, USA

Joint work with Christopher Ambrose, Joppe W. Bos, Björn Fay, Manfred Lochter, and Bruce Murray

Elliptic Curve Signatures

1985	ElGamal introduces first DLog-based signatures
1985	Koblitz and Miller propose elliptic curve cryptography
(?) 1991	Kravitz designs a variant of ElGamal signature scheme (DSA)
1993	NIST adopts DSA as FIPS 186
2000	NIST includes ECDSA in FIPS 186-2
2012	Bernstein, Duif, Lange, Schwabe, and Yang publish EdDSA


Approved American National Standard

ANSI

American National Standard
for Financial Services

ANS X9.62–2005

Public Key Cryptography for the Financial Services Industry

The Elliptic Curve Digital Signature Algorithm (ECDSA)

X9

Accredited Standards Committee X9, Inc.

Financial Industry Standards

Date Approved: November 16, 2005

American National Standards Institute

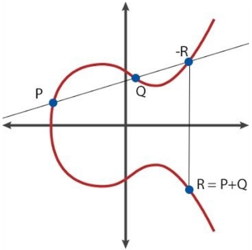
ECDSA

Elliptic Curve Digital Signature Algorithm

Key generation

public key $\langle G \rangle$ of prime order n

private key $d \xleftarrow{\$} [1, n-1]$



Signature

Input: Message $m \Rightarrow e = \mathcal{H}(m)$

1. $k \xleftarrow{\$} [1, n-1]$
2. $r \leftarrow \mathrm{x}([k]G) \bmod n$ $\quad \triangleright\, r \neq 0$
3. $s \leftarrow k^{-1}(e + dr) \bmod n$ $\quad \triangleright\, s \neq 0$

Output: Signature on m is (r, s)

random integer k cannot be re-used!!

Sensitivity

ECDSA is sensitive to PRNG's quality

- nonce k cannot be re-used
- worse, prediction of a number of bits of k allows the recovery of private key d

Given signatures (r_1, s_1) and (r_2, s_2) on 2 different messages m_1 and m_2:

$$\begin{cases} s_1 \leftarrow k_1^{-1}(e_1 + dr_1) \bmod n \\ s_2 \leftarrow k_2^{-1}(e_2 + dr_2) \bmod n \end{cases}$$

If $k_1 = k_2$ then $d = \ldots$

Dec 2010: `Fail0verflow` recovers ECDSA private key used to sign code for PS3

Solution

Generate signatures in a completely deterministic way

Deterministic ECDSA

ECDSA

Input: Message $m \Rightarrow e = \mathcal{H}(m)$

1. $k \xleftarrow{\$} [1, n-1]$
2. $r \leftarrow \mathrm{x}([k]G) \bmod n$
3. $s \leftarrow k^{-1}(e + dr) \bmod n$

Output: Signature on m is (r, s)

Deterministic ECDSA

Input: Message $m \Rightarrow e = \mathcal{H}(m)$

1. $u \leftarrow \mathrm{GenU}(d, e)$
2. $r \leftarrow \mathrm{x}([u]G) \bmod n$
3. $s \leftarrow u^{-1}(e + dr) \bmod n$

Output: Signature on m is (r, s)

This is the approach used in EdDSA

Edwards-curve DSA
Bernstein, Duif, Lange, Schwabe, and Yang
JCEN, 2012

Solution

Generate signatures in a completely deterministic way

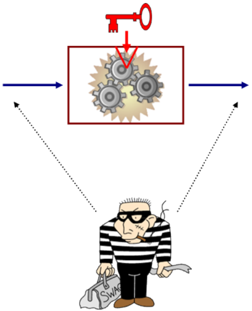
Question

Is it always more secure?

Physical Attacks

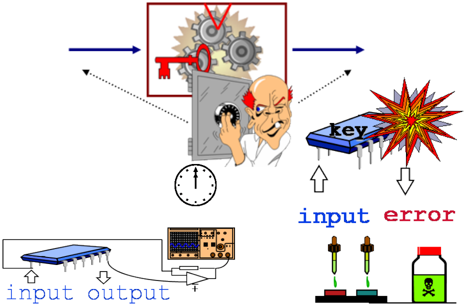
Black-box attacks

- Attacker has only access to I/O



Physical attacks

- Attacker can monitor the execution and inject faults



Edwards-Curve DSA (EdDSA)

Key generation

public key $\langle B \rangle$ of prime order ℓ and $\underline{A}$
where $A = sB$

private key $k \xleftarrow{\$} [1, \ell]$
$\rightsquigarrow \mathcal{H}_2(k) = (h_0, h_1, \ldots, h_{2b-1})$
$\rightsquigarrow s = 2^c \cdot (1, h_{n-1}, \ldots, h_c)_2$

Signature

Input: Message m

1. $m' \leftarrow \mathcal{H}'(m)$ $\triangleright$ prehash function $\mathcal{H}'$
2. $r \leftarrow \mathcal{H}(h_b, \ldots, h_{2b-1}, m') \bmod \ell$
3. $R \leftarrow rB$
4. $t \leftarrow \mathcal{H}(\underline{R}, \underline{A}, m')$
5. $S \leftarrow r + ts \bmod \ell$

Output: Signature on m is $(\underline{R}, \underline{S})$

Attacks against EdDSA

Fault attack on base-point B during import: $B \rightsquigarrow \tilde{B}$

Secret s can be recovered since

$$S - \tilde{S} \equiv (t - \tilde{t})s \pmod{\ell}$$

and t, $\tilde{t}$ can be computed

Signature

Input: Message m

1. $m' \leftarrow \mathcal{H}'(m)$
2. $r \leftarrow \mathcal{H}(h_b, \ldots, h_{2b-1}, m') \bmod \ell$
3. $\tilde{R} \leftarrow r\tilde{B}$
4. $\tilde{t} \leftarrow \mathcal{H}(\underline{\tilde{R}}, \underline{A}, m')$
5. $\tilde{S} \leftarrow r + \tilde{t}s \bmod \ell$

Output: Signature on m is $(\underline{\tilde{R}}, \underline{\tilde{S}})$

Our Results

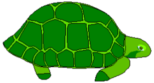
where	attack	type	number of faults
Import point B	fault	uncontrolled	≥ 1
Import point A	fault	controlled	≥ 1
Hash computation of r	fault	controlled	≥ 1
Hash computation of r with fixed (unknown) output	$\{$ fault	uncontrolled	≥ 1 $\}$
Scalar multiplication rB	fault	uncontrolled	≥ 1
Hash computation of t	fault	controlled	≥ 1
Hash computation of t with fixed (unknown) output	$\{$ fault	controlled	≥ 2 $\}$
Computation of S	fault	controlled	≥ 1
Hash computation of r	DPA/DEMA	–	–

Countermeasures (1)

Fully compliant countermeasures

- Check the validity of targeted points ▷ do not cover all our attacks!
- Use redundancy (e.g., double computation)
- Harden the hash computation ▷ against the side-channel attack

⇝ Significant impact on performance



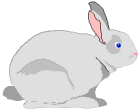
Countermeasures (2)

Not fully compliant countermeasures

- Adaptive solution $\rightsquigarrow$ Include random noise in the computation of r

$$r \leftarrow \mathcal{H}(\underbrace{\kappa}_{\text{random noise}}, \underbrace{h_b, \ldots, h_{2b-1}}_{\text{secret input}}, \underbrace{m'}_{\text{prehashed message } m'}) \bmod \ell$$

(or unknown counter if no random source is available)



Summary

- Removing randomness in signature generation does not necessarily eliminate all attack vectors: 8 fault attacks and 1 side-channel attack
- Countermeasures fully compliant with the current specification of EdDSA seem to have a significant performance impact
- Deviating from the specification and introducing high-quality randomness [where this is possible] allows the construction of cheap countermeasures
  - and does not affect the key generation and signature verification

We hope this work serves as valuable input when the community and the various standardization bodies start to define new cryptographic digital signature algorithms

Comments/Questions?