Provable Security in Cryptography

DL-based Systems

ECC - Sept 24th 2002 - Essen

David Pointcheval
Ecole normale supérieure
France

Summary

- The Methodology of "Provable Security"
- Complexity Assumptions
- Encryption
- Signature
- Conclusions

Provable Security in Cryptography

DL-based Systems

Provable Security

David Pointcheval
Ecole normale supérieure
France

Provable Security: a Short Story

- Originated in the late 80's
  - encryption [GM86]
  - signature [GMR88]
- Increased applicability using ideal substitutes
  - random oracles vs hash functions [FS86, BR93]
  - generic groups vs elliptic curves [Na94,Sh97]
  - ideal ciphers vs block ciphers [BPR EC'00]
- Now requested to support emerging standards (IEEE P1363, ISO, Cryptrec, NESSIE)

The Need for Provable Security

- "Textbook" cryptosystems cannot be used as such
  (homomorphic properties, …)
- Practitioners need formatting rules to ensure interoperability

⇒ Paddings are used in practice: heuristic

- PKCS#1 V 1.5 - Encrypt [Bl98]
- PKCS#1 V 2.0 - Encrypt [Ma01]
- ISO 9796-1 - Signature [CNS99, CHJ99]

Attacks

The Limits of Provable Security

- Provable security does not yield proofs
  - proofs are relative (to computational assumptions)
  - proofs often use ideal models (ROM, ICM, GM)
    Meaning is debatable
    - ROM [CGH98]
    - GM [SPMS C'02]
  - proofs are not formal objects
    Time is needed for acceptance.
- Still, provable security is a means to provide some form of guarantee that a scheme is not flawed

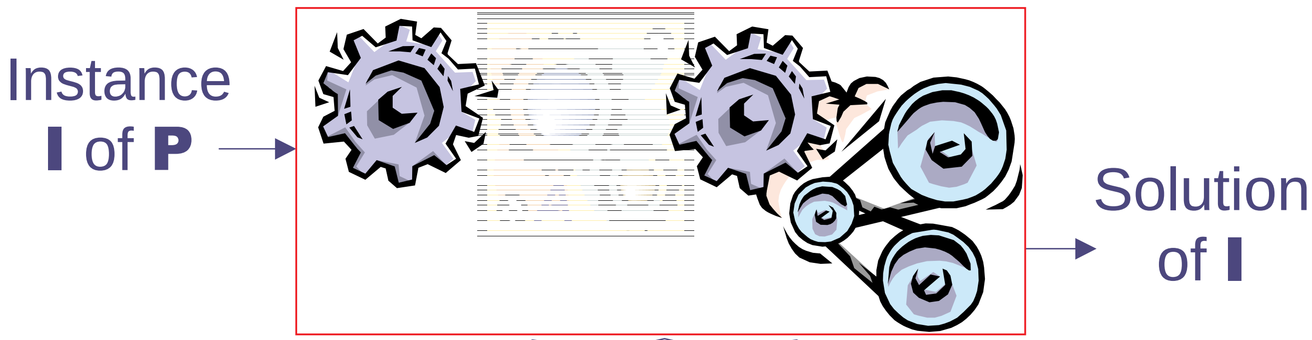
Provable Security

1 - Define goal of adversary
2 - Define security model
3 - Define complexity assumptions
4 - Provide a proof by reduction
5 - Check proof
6 - Interpret proof

Proof by Reduction

Reduction of a problem **P** to an attack *Atk*:

- Let *A* be an adversary that breaks the scheme then *A* can be used to solve **P**



P intractable ⇒ scheme unbreakable

Provable Security in Cryptography

DL-based Systems

Assumptions

David Pointcheval
Ecole normale supérieure
France

Integer Factoring and RSA

- Multiplication/Factorization :
  - $p, q \mapsto n = p.q$ easy (quadratic)
  - $n = p.q \mapsto p, q$ difficult (super-polynomial)

One-Way Function

- RSA Function, from $\mathbb{Z}_n$ in $\mathbb{Z}_n$ (with $n{=}pq$) for a fixed exponent e

Rivest-Shamir-Adleman '78

  - $x \mapsto x^e \bmod n$ easy (cubic)
  - $y{=}x^e \bmod n \mapsto x$ difficult (without p or q)

RSA Problem

$x = y^d \bmod n$ where $d = e^{-1} \bmod \varphi(n)$

trapdoor

$$\mathrm{Succ}^{\mathrm{rsa}}_{n,e}(A) = \Pr_{x \in \mathbb{Z}_n^*}\left[A(y) = x \,\middle|\, y = x^e \bmod n\right]$$

The Discrete Logarithm

- Let **G** = (<g>, ×) be any finite cyclic group
- For any $y \in$ **G**, one defines
  $$\mathrm{Log}_g(y) = \min\{x \geq 0 \mid y = g^x\}$$
- One-way function
  - $x \rightarrow y = g^x$ easy (cubic)
  - $y = g^x \rightarrow x$ difficult (super-polynomial)

$$\mathrm{Succ}_g^{\mathrm{dl}}(\mathcal{A}) = \Pr_{x \in \mathbf{Z}_q}\left[\mathcal{A}(y) = x \,\middle|\, y = g^x\right]$$

Any Trapdoor …?

- The Discrete Logarithm is difficult
  and no information could help!
- The Diffie-Hellman Problem (1976):
  - Given $A=g^a$ and $B=g^b$
  - Compute DH(A,B) = $C=g^{ab}$

Clearly CDH ≤ DL: with $a=\mathrm{Log}_g A$, $C=B^a$

$$\mathrm{Succ}_g^{\mathrm{cdh}}(\mathcal{A}) = \Pr_{a,b \in \mathbf{Z}_q}\left[\mathcal{A}(A,B) = C \,\middle|\, A = g^a, B = g^b, C = g^{ab}\right]$$

Other DL-based Problems

The **Decisional Diffie-Hellman Problem**:

- Given A, B and C in $<g>$
- Decide whether C = DH(A,B)

The **Gap Diffie-Hellman Problem**:

Okamoto-Pointcheval PKC'01

Solve the computational problem, with access to a decisional oracle

Weak curves: DDH is easy, because of pairing, then GDH=CDH

Complexity Estimates

Estimates for integer factoring [LV PKC'00]

	Modulus (bits)	Mips-Year ($\log_2$)	Operations (en $\log_2$)
	512	13	58
Mile-stone	1024	35	80
	2048	66	111
	4096	104	149
	8192	156	201

Can be used for RSA too

Lower-bounds for DL in $\mathbb{Z}_p^*$

Provable Security in Cryptography

DL-based Systems

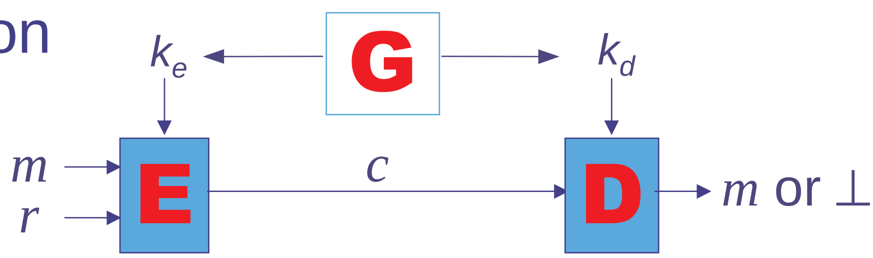
Encryption

David Pointcheval
Ecole normale supérieure
France

Encryption Scheme

3 algorithms :

- **G** - key generation
- **E** - encryption
- **D** - decryption



OW-Security: it is impossible to get back m just from c, k_e, **E** and **D** (without k_d)

Weaker Goals of Adversary

- Perfect Secrecy:

  the ciphertext and public data do not reveal any information about the plaintext (but maybe the size)

  Information Theoretical sense ⇒ Impossible

- Semantic Security (Indistinguishability):

  IND

  no polynomial adversary can learn any information about the plaintext from the ciphertext and public data (but the size)

Security Models

- Chosen Plaintext: *(basic scenario)*

  CPA

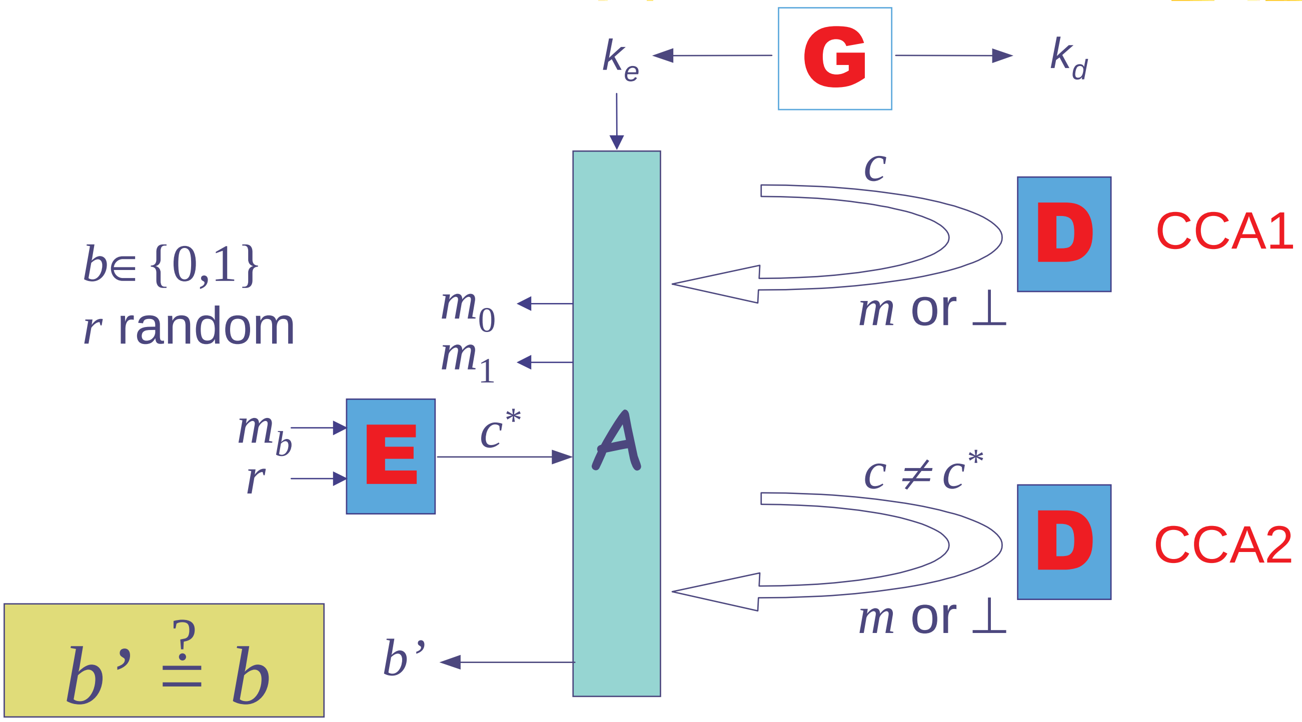
  in the public-key setting, any adversary can get the encryption of any plaintext of his choice (by encrypting it by himself)

- Chosen Ciphertext (adaptively):

  CCA2

  the adversary has furthermore access to a decryption oracle which decrypts any ciphertext of his choice, but the specific challenge

IND-CCA2



Main Security Notions

- IND-CCA2: *(the strongest - [BDPR C'98])*

$$2 \Pr_{r,b}\left[A_2^{\mathbf{D}}(m_0, m_1, c, s) = b \;\middle|\; \begin{matrix} (m_0, m_1, s) \leftarrow A_1^{\mathbf{D}}(k_e) \\ c \leftarrow \mathbf{E}(m_b, r) \end{matrix} \right] - 1$$

= Advantage negligible

- OW-CPA: *(the weakest)*

$$\Pr_{m,r}\left[A(c) = m \;\middle|\; c = \mathbf{E}(m;r) \right]$$

= Success negligible

Practical Cryptosystems

- Integer Factoring-based: RSA [RSA78]
  - OW-CPA = RSA (modular e-th roots)
  - IND ? No, because of determinism
  - CCA2 ? No, because of multiplicativity
- DL-based: El Gamal [EG85]
  - OW-CPA = CDH
  - IND-CPA = DDH
  - CCA2 ? No, because of multiplicativity

Generic Conversions

- Any trapdoor one-way function leads to a OW-CPA cryptosystem
- But OW-CPA not enough
- How to reach IND-CCA2 ?

  $\Rightarrow$ generic conversions
  from weakly secure schemes
  to strongly secure cryptosystems

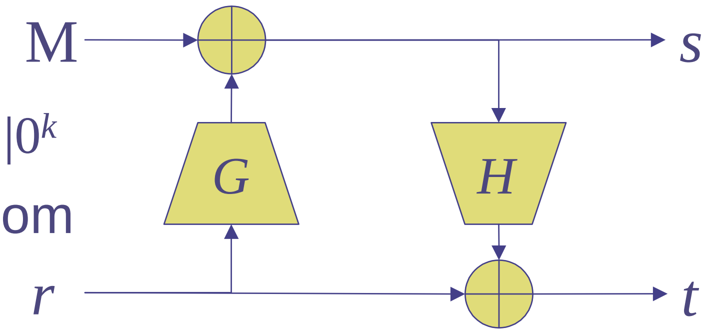
OAEP

Bellare-Rogaway EC'94

Let f be a trapdoor one-way permutation,
with $G \to \{0,1\}^n$ and $H \to \{0,1\}^\ell$

$\mathrm{M} = m||0^k$

r random



E(m,r) : Compute s,t then return $c=f(s||t)$
D(c) : Compute $s||t = f^{-1}(c)$, invert OAEP,
then check redundancy

OAEP: Security Level

In 1994, Bellare and Rogaway proved that

- the OAEP construction provides an IND-CPA cryptosystem under the OW of f
- it is plaintext-aware (PA94)

Widely believed: IND-CPA + PA94 $\Rightarrow$ IND-CCA2

But IND-CPA + PA94 $\Rightarrow$ IND-CCA1 only

We improved PA94 into PA98 [BDPR C'98]

IND-CPA + PA98 $\Rightarrow$ IND-CCA2

But… PA98 of OAEP never studied

OAEP: Security Level

Until 2000, OAEP was anyway believed to provide an IND-CCA2 cryptosystem under the OW of f

But Shoup showed a counter-example [Sh C'01]

A stronger assumption about f is required: under the partial-domain OW of f, OAEP provides an IND-CCA2 cryptosystem [FOPS C'01]

OW: $f(x) \rightarrow x$ hard   PD-OW: $f(x,y) \rightarrow x$ hard

RSA-OAEP: Interpretation

$$\mathrm{Adv}^{ind}(t) \leq 2 \times \sqrt{\mathrm{Succ}^{\mathrm{rsa}}_{n,e}\left(2t + q_H\left(2q_G + q_H\right)k^3\right)}$$

Security bound: 2^{75}, and 2^{55} hash queries

If one can break the scheme within time T, one can invert RSA within time $T' \leq 2\,T + 2\,q_H(2q_G + q_H)\,k^3$

$\leq 2 \times 2^{75} + 6 \times 2^{110}\ k^3 < 2^{113}\,k^3$

modulus: 1024 bits $\rightarrow 2^{143}$ (NFS: 2^{80}) ✗

2048 bits $\rightarrow 2^{146}$ (NFS: 2^{111}) ✗

4096 bits $\rightarrow 2^{149}$ (NFS: 2^{149}) ✓

REACT

Okamoto-Pointcheval RSA'01

Let f be an injective function,
which provides a ***Gap-Problem***:

OW even given access to a checking oracle
(on input (x,y) answers whether $x = f^{-1}(y)$)

E$(m\ ; r) = (a, b, c)$ with $a = f(r)$, $b = \mathrm{E}_{G(r)}(m)$
and $c = H(m,r,a,b)$

D(a,b,c): compute $r = f^{-1}(a)$ and $m = \mathrm{D}_{G(r)}(b)$
if $c = \mathrm{H}(m,r,a,b)$ then output m
otherwise: $\perp$ (*reject*)

REACT: Security Result

$$\mathrm{Adv}^{ind}(t) \leq \mathrm{Adv}_{\mathsf{E}}^{ind}(t) + 2\mathrm{Succ}_f^{gap}(t, q_G + q_H) + \frac{4q_{\mathsf{D}}}{2^k}$$

Security bound: 2^{75}, and 2^{55} hash queries

If one can break the scheme
within time T, one can invert f within time
$T' \leq T + (q_G + q_H)\, T_{check} \leq T + 2^{55}\, T_{check}$

RSA small exponent: 1024 bits $\rightarrow$ Secure

ElGamal: GDH $\rightarrow$ 160 bit order group

PSEC-3 = REACT-EC-ElGamal

REACT-EC-EG ≈ ECIES ABR RSA'01

- G a MGF, M a MAC
- E, D: symmetric encryption scheme

E$(m\ ;\ r)$: $(\mathbf{A}, B, C)$
where $\mathbf{A} \leftarrow r.\mathbf{P}$,
$\mathbf{K} \leftarrow r.\mathbf{Y}$, $k \leftarrow G(\mathbf{K})$,
$B \leftarrow \mathrm{E}_k(m)$, $C \leftarrow \mathrm{M}_k(B)$

x : **secret** key
$\mathbf{Y} = x.\mathbf{P}$: **public** key

D$(\mathbf{A},B,C)$: $\mathbf{K} \leftarrow x.\mathbf{A}$,
$k \leftarrow G(\mathbf{K})$, $m \leftarrow \mathrm{D}_k(B)$,
check whether $C = \mathrm{M}_k(B)$

ECIES: Security Result

Theoretical security result (from ABR):
- relative to ODH assumption
- or GDH + ROM (similar to REACT-EC-EG)

But in SEC1 description (Certicom)

$r \leftarrow_R \mathbb{Z}_q$, $\mathbf{A} \leftarrow r.\mathbf{P}$, $\mathbf{K} \leftarrow r.\mathbf{Y}$, $k \leftarrow G(\mathbf{K})$
modified into $k \leftarrow G(\mathbf{K}_x)$

D$(\mathbf{A}, B, C)$ = **D**$(-\mathbf{A}, B, C)$: malleability!

- Not a real security concern, gCCA2 model
  Problem = *partial* encoding $\mathbf{K}_x$ of $\mathbf{K}$

Provable Security in Cryptography

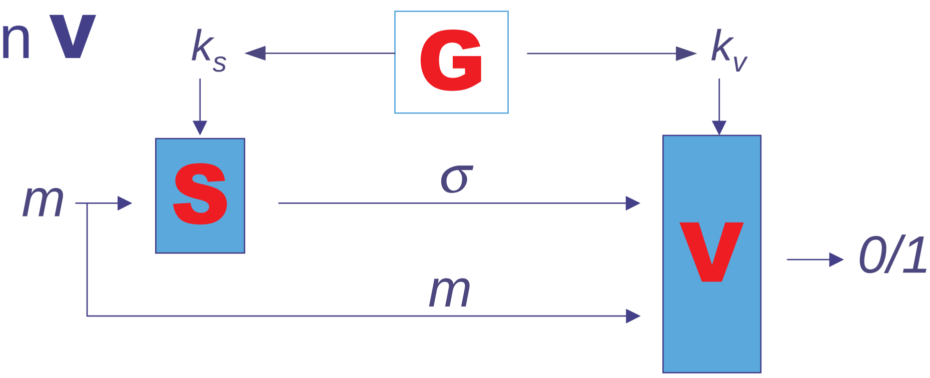
DL-based Systems

Signature

David Pointcheval
Ecole normale supérieure
France

Signature Scheme

- Key Generation **G**
- Signature **S**
- Verification **V**



Non-repudiation: impossible
to forge valid σ without k_s

Goal of the Adversary

- Existential Forgery:

  Try to forge a valid message-signature pair without the private key

  Adversary is successful if the following probability is large

$$\mathrm{Succ}^{ef}(\mathcal{A}) = \Pr\left[\mathbf{V}(m,\sigma) = 1 \middle| \mathcal{A}(k_v) = (m,\sigma)\right]$$

Security Models

- No-Message Attacks: the adversary only knows the verification (public) key
- Known-Message Attacks (KMA): the adversary has access to a list Λ of message/signature pairs
- Chosen-Message Attacks (CMA): the messages are adaptively chosen by the adversary

  $\Rightarrow$ the strongest attack

Probabilistic Signatures - 1

- In a probabilistic signature scheme, several signatures may correspond to a message
- In the usual definition for
  Chosen-Message Attacks (CMA), the adversary can repeatedly submit a same message.

Otherwise, weaker model :

- Single-Occurrence Chosen-Message Attacks (SO-CMA) - each message m can be submitted only once

ESIGN

Fujioka-Okamoto-Miyaguchi EC'91

A signature scheme designed in the early 90ies and considered in IEEE P1363, Cryptrec NESSIE, together with a security proof

- Proof holds only in SO-CMA scenario
- Interpretation:
  – ESIGN is not broken, but not provably UF-CMA
  – either give up CMA property…
  – or tweak ESIGN

Probabilistic Signatures - 2

- In the usual definition for
  Existential Forgery, output forgery corresponds to a fresh message m.
  No pair $(m\ \sigma)$ can be in the list Λ.

Otherwise, weaker goal:

- Malleability: produce a new pair $(m,\sigma) \notin \Lambda$ possibly for a submitted message m.
  $((m,\sigma')$ in Λ for some $\sigma' \neq \sigma)$
- Non-malleability is a *stronger demand* than resistance to existential forgeries

Schnorr Signature

Schnorr EC '89

G, g and q: **common** elements

x: **private** key $y=g^x$: **public** key

Signing m:

choose $k \in \mathbf{Z}_q$ and compute $r=g^k$

as well as $e=H(m,r)$

and $s = k\text{-}xe \bmod q$

$\sigma = (e,s)$

Verifying (m,σ):

$u = g^s y^e \ (= g^{k-xe} g^{xe})$

test if $e=H(m,u)$

Security Proof Pointcheval-Stern EC'96

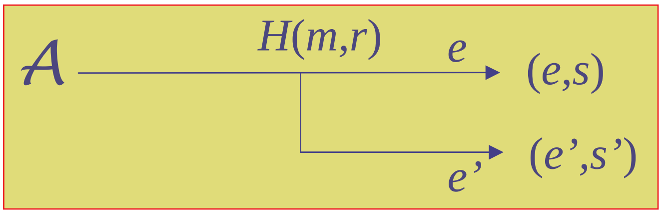
Existential Forgery = DL problem

Idea : *forking lemma*

Run $\mathcal{A}$ once

In case of success:

run $\mathcal{A}$ again



One gets two successes with probability $\geq \varepsilon^2 / 4\, q_H$

Improvement:

two successes in q_H / ε expected iterations

$g^s y^e = r = g^{s'} y^{e'}$

$g^{s-s'} = y^{e'-e}$

Let $\alpha = (s-s')/(e'-e) \bmod q$

Then $y = g^\alpha$

Comments: Forking Lemma

Security bound: 2^{75}, and 2^{55} hash queries

If one can break the scheme within time $T = t/\varepsilon$, one can extract two tuples within time

$$T' \leq q_H\, t/\varepsilon = q_H\, T \leq 2^{130}$$

This is not a practical result:

- 4096 bit moduli are required in $\mathbf{Z}_p^*$
- 260 bit order are required in EC

ECDSA

G $= \langle \mathbf{P} \rangle$, $\mathbf{P}$ an element of order q of **EC**,
x: **private** key $\mathbf{Y} = x.\mathbf{P}$: **public** key

$\sigma = (r,s)$

Signing m:

- choose $k \in \mathbb{Z}_q$
- compute $\mathbf{R} = k.\mathbf{P}$
- compute $r = f(\mathbf{R})$
- compute $e = H(m)$, $s = (e + xr)/k \bmod q$

Verifying (m,r,s): first $0 < r, s < q$

- compute $\mathbf{R'} = e\, s^{-1}.\mathbf{P} + r\, s^{-1}.\mathbf{Y}$

test if $r = f(\mathbf{R'})$

ECDSA: Security Result

Brown '00

- With almost-invertible functions f

In the Generic Model, non-malleability of ECDSA cannot be broken with probability significantly greater than $5(n+1)(n+q_s+1)/q$

q_s # of signing queries - n # of group operations

In ECDSA, $f(\mathbf{R}) = \mathit{first\text{-}coordinate}(\mathbf{R}) = x_{\mathbf{R}}$,
which is an almost-invertible function

$\Rightarrow$ In the Generic Model, ECDSA is NM

ECDSA: Malleability

- In ECDSA, $f(\mathbf{R}) = \textit{first-coordinate}(\mathbf{R}) = x_{\mathbf{R}}$
  Thus $f(-\mathbf{R}) = f(\mathbf{R})$
  Given a valid signature (m,r,s),
  one obtains another as $(m,r,-s \bmod q)$
  This is exactly malleability
- Interpretation:
  - ECDSA is not broken (provides non-repudiation)
    problem = *partial* encoding (again!)
  - to eliminate malleability need to tweak ECDSA

ECDSA: Interpretation

- The security proof "proves" a property that **does not hold** for the actual scheme
- Interpretation:
  - **EC** groups are **not generic**
    (they have automorphisms)
  - either change the model…
  - or tweak the scheme

Provable Security in Cryptography

DL-based Systems

Conclusion

David Pointcheval
Ecole normale supérieure
France

Ideal Models

Ideal models to be handled with care

- Random oracle model:
  seems correct in practice
  still not a security proof
  but a security argument
- Generic model: less convincing
  still better than nothing.

  This model could be improved:
  taking care of automorphisms.

Provable Security

Well studied / Almost done: 1 - Define goal of adversary; 2 - Define security model — Security notions

1 - Define goal of adversary

2 - Define security model

Security notions

Algorithmic: 3 - Define complexity assumptions

Classical: 4 - Provide a proof by reduction

Difficult: 5 - Check proof ← *Shoup's methodology makes it easier*

6 - Interpret proof

Very few proofs are meaningful in pratice…

- proofs to be improved?
- schemes to be modified?