A Conditional Security Analysis
of the Elliptic Curve
Digital Signature Algorithm

Daniel R. L. Brown

𝔼CC 2001

University of Waterloo
3:30pm–4:30pm, Tuesday, October 30, 2001

certicom

Outline

1. **Introduction:** Standardization, Signatures Schemes, ECDSA, and DSA
2. **Security of Signature Schemes:** How to define forgery.
3. **Necessary Conditions for Security of (EC)DSA:** Some Flaws that Must be Avoided.
4. **Introduction to Provable Security:** Reductions to Hard Problems as Security Proofs
5. **Additional Conditions for the Proofs:** *Necessary* Conditions for Security of (EC)DSA, when combined with sufficient conditions.
6. **Security Proofs:** Five Sketches of Reductions
7. **Summary, Remarks and Conclusion**

Introduction

1. **Standardization of ECDSA**
2. **Signature Schemes**
3. **Common Features of ECDSA**
4. **Diagrams**
5. **Effective Hash Function of ECDSA and DSA**
6. **Difference between ECDSA and DSA**

Development and Standardization of ECDSA

Academic proposals for elliptic curve based digital signatures were made in 1985*. Standardization efforts towards an elliptic curve analogue of DSA, the Elliptic Curve Digital Signature Algorithm (ECDSA), began in 1992†.

Approved ECDSA Standards:

1999 Jan.	ANSI X9.62
2000 Jan.	FIPS 186–2
2000 Aug.	IEEE Std 1363–2000

*Koblitz, Miller
†Vanstone, Menezes, . . .

Definition of a Signature Scheme

A signature scheme is a triple of probabilistic algorithms (KeyGen, Sign, Verify) of the following form.

$$\mathtt{KeyGen}() = (d, Q) \quad (1)$$

$$\mathtt{Sign}(d, M) = \sigma \quad (2)$$

$$\mathtt{Verify}(Q, M, \sigma) = v \quad (3)$$

where

d	is a Private key
Q	is a Public key
M	is a Message
σ	is a Signature
v	is a Rejection or Acceptance

If all three equations (1)–(3) hold then v must be acceptance.

Features Shared by ECDSA and DSA

Both ECDSA and DSA signing work as follows.

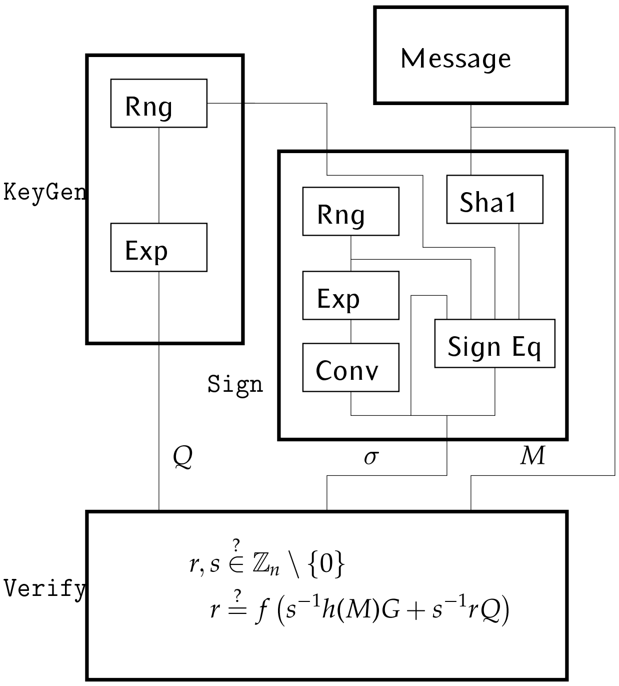
KeyGen:	Sign:	
$d \xleftarrow{\$} \mathbb{Z}_n \setminus \{0\}$	$k \xleftarrow{\$} \mathbb{Z}_n \setminus \{0\}$	
$Q \leftarrow dG$	$r \leftarrow f(kG)$	$(\neq 0)$
	$e \leftarrow h(M)$	
	$s \leftarrow k^{-1}(e + rd)$	$(\neq 0)$
	$\sigma \leftarrow (r, s)$	

with notation as follows:

$\xleftarrow{\$}$	Selection of a random element
n	Prime number larger than 2^{159}
$\mathbb{Z}_n$	Integers modulo n
G	Generator of a group $\langle G \rangle$ of order n
dG, kG	Scalar multiplication*in $\langle G \rangle$
$h : \{0,1\}^* \to \mathbb{Z}_n$	Hash function (note range)
$e \in \mathbb{Z}_n$	Hash value
$k \in \mathbb{Z}_n$	One-time private key
$f : \langle G \rangle \to \mathbb{Z}_n$	**Conversion** function
$r \in \mathbb{Z}_n$	First signature part
$s \in \mathbb{Z}_n$	Second signature part

*Wlog, use additive notation for $\langle G \rangle$.

Diagrams and Verification



Effective Hash Function

The effective* hash function h used by both DSA and ECDSA is:

$$h : \{0,1\}^* \to \mathbb{Z}_n : M \mapsto (\overline{\mathrm{Sha1}(M)} \mod n)$$

where Sha1 : $\{0,1\}^* \to \{0,1\}^{160}$ is the FIPS 180-1 secure hash algorithm, and $\overline{\mathrm{Sha1}(M)}$ means the result of converting the bit string in $\{0,1\}^{160}$ to an integer.

Future editions of ECDSA and DSA may be based on the new hash functions, SHA256, SHA384, and SHA512 for greater security.

*We consider h (not just Sha1) as the effective hash function, for the purposes of security analysis. We will see it is h that affects security, not just Sha1, thus the term effective.

Differences Between ECDSA and DSA

ECDSA and DSA differ mainly in their choice of group and conversion function. They both use a **supergroup**, a group containing the group $\langle G \rangle$. The index t of $\langle G \rangle$ in the supergroup is the **cofactor**.

	ECDSA	DSA
Supergroup:	$\mathbb{E}(\mathbb{F}_q)$	$\mathbb{F}_p^*$
Conversion f:	$(x, y) \mapsto (\bar{x} \bmod n)$	$x \mapsto (x \bmod n)$
Order n:	$n \approx 2^m/t$	$2^{159} \leq n \leq 2^{160}$
Cofactor t:	$t = 1, 2, 4$	$t = (p-1)/n \approx 2^{864}$

In the case of ECDSA, $x \in \mathbb{F}_q$ where $q \approx 2^m$ is a prime power field order and $m \in \mathcal{M}^*$. Also, $\bar{x}$ means conversion of the field representation of x to an integer.

*For NIST, $\mathcal{M} = \{163, 192, 224, 233, 256, 283, 384, 409, 521, 571\}$

Security of Signature Schemes

1. **Fundamental Types of Forgers**
2. **Relationships between the Types of Forgers**
3. **Diagrams**

Fundamental† Types of Forger

An *existential forger launching an adaptive chosen-message attack*‡ `Forge` works as follows.

Input: A public key Q where $(d, Q) \leftarrow \texttt{KeyGen}()$.
Help: A signing oracle yielding signatures σ_i that `Forge` can invoke for chosen query messages M_i, ie $\sigma_i \leftarrow \texttt{Sign}(d, M_i)$.
Output: A forged message M and signature σ such that $M \neq M_i$ for all i and $\texttt{Verify}(Q, M, \sigma)$ is Accept.

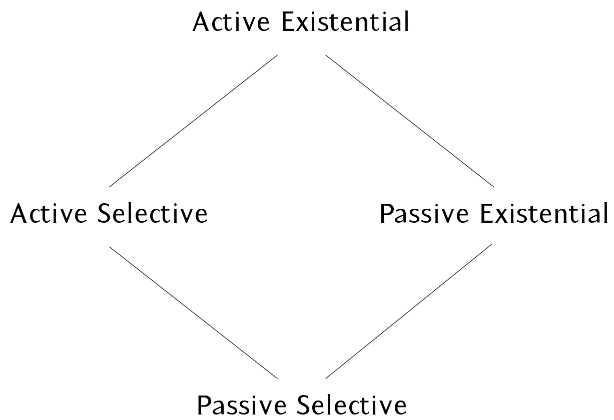
A *no-message attack* does not invoke the signing oracle.

A *selective forger* takes $M \xleftarrow{\$} \mathcal{S}$ as an input, from some distribution $\mathcal{S}$.

†We won't consider non-fundamental forgers and other adversaries such as: forged additional signatures on a message, forged additional public key for a signature, extraction the public key from message and signature only, and misauthentication in the multi-user setting.

‡Goldwasser, Micali, Rivest

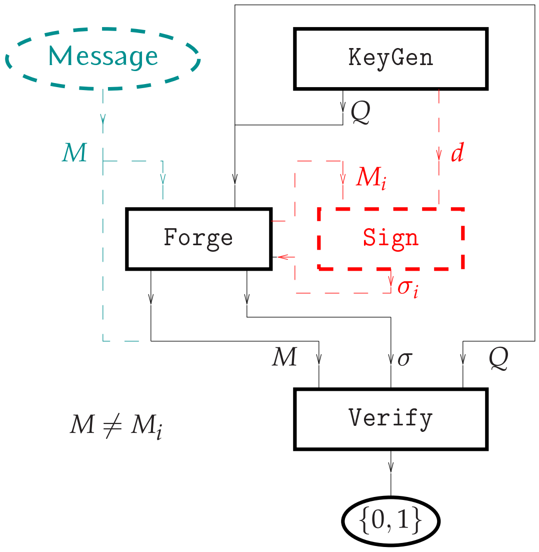
Relations between Fundamental Types of Forger



Some abbreviations:

Passive	= No-Message Attack
Active	= Adaptive Chosen-Message Attack

Forgers: Selective/Existential and Passive/Active



Conditions Necessary for the Security of (EC)DSA

1. **Random Number Generator**
2. **Hash Function**
3. **Conversion Function**
4. **Group**
5. **Range Check**
6. **Summary**

Flaw To Avoid 1: Weak RNG

The following flaws in the one-time private key k generator allow active selective forgery because the private key d is revealed by information leaked in s.

- Repeated one-time keys.
- Linear congruential RNGs*.
- Leakage of bits in the binary representation†.
- Slight arithmetic bias.‡

*Bellare, Goldwasser, Micciancio
†Smart, Howgrave-Graham, Nguyen, Shparlinski, ...
‡Bleichenbacher

Flaw To Avoid 2: Weak Hash Function

- Collision-finder*† $\Longrightarrow$ Active Existential Forger
- 2nd-preimage-finder $\Longrightarrow$ Active Selective Forger
- Preimage-finder $\Longrightarrow$ Passive Existential Forger
  $x, y \xleftarrow{\$} \mathbb{Z}_n \setminus \{0\}$
  $r \leftarrow f(xG + yQ)$
  Solve $s^{-1}r = y$ for s
  Solve $s^{-1}e = x$ for e
  Invert h at e
- Zero-finder‡§ $\Longrightarrow$ Passive Existential Forger¶
- Frequently-zero $\Longrightarrow$ Passive Selective Forger

*Simplification of Damgård
†Domain parameter attack (Vaudenay): $n \leftarrow$ Sha1(M_0) − Sha1(M_1)
‡Two zeroes $\Longrightarrow$ collision
§Domain parameter attack: $n \leftarrow$ Sha1(M_0)
¶Independently: Kaliski, Wang, Anonymous referee

Flaw To Avoid 3: Weak Conversion Function

- Very large f-preimage $\implies$ Passive Selective Forger*.
  - Suppose $|f^{-1}(r^*)| \approx n/700$
  - $r \leftarrow r^*$.
  - $s \xleftarrow{\$} \mathbb{Z}_n \setminus \{0\}$
  - (r,s) will verify with probability about $1/700$, since verification checks that $r \stackrel{?}{=} f\left(s^{-1}(h(M)G + rQ)\right)$ and s randomizes the input to f over $\langle G \rangle$.
- DSA: partial results to avoiding this problem.† ECDSA: easily avoids this problem.
- This weakness is easily detected, though.
- If f has no large preimage then it is *almost-bijective*

*Anonymous referee

†Shparlinksi, Nguyen, ...

Flaw To Avoid 4: Weak Group (and Conversion)

- Discrete-Log-Finder $\implies$ Passive Selective Forger
- Semilog-Finder $\implies$ Passive Selective Forger
  - $(t,u) \in \mathbb{Z}_n^2$ is a **semilogarithm** of P to the base G if $t = f\left(u^{-1}(G + tP)\right)$.
  - Forgery on M under public key Q:

    $P \leftarrow h(M)^{-1}Q$
    Find semilogarithm (t,u) of P.
    $r \leftarrow t$
    $s \leftarrow h(M)u$
    Then $r = t = f\left(u^{-1}(G + tP)\right)$
    $= f\left(s^{-1}h(M)(G + tP)\right)$
    $= f\left(s^{-1}(h(M)G + rQ)\right)$
- Semilog-Resistant $\implies$ Almost-Bijective f.

Flaw To Avoid 5: Allowing $r = 0$ in Verification

- If $f(tG) = 0$ with t known*† $\implies$ Passive Selective Forger‡
  - $r \leftarrow 0$
  - $s \leftarrow t^{-1}h(M)$
  - $r = 0 = f(tG) = f\left(th(M)^{-1}h(M)G + \mathcal{O}\right)$
    $= f\left(s^{-1}h(M)G + s^{-1}rQ\right)$

*Domain parameter attack on DSA: search for (p,q,g) such that $ord_p(g) = q$ and $((g^t \bmod p) \bmod q) = 0$

†Domain parameter attack on ECDSA: search for $\mathbb{E}$ of order hn with point $G = (x,y)$ of order n where $\bar{x} = 0 \mod n$

‡Anonymous referee. Johnson, Menezes

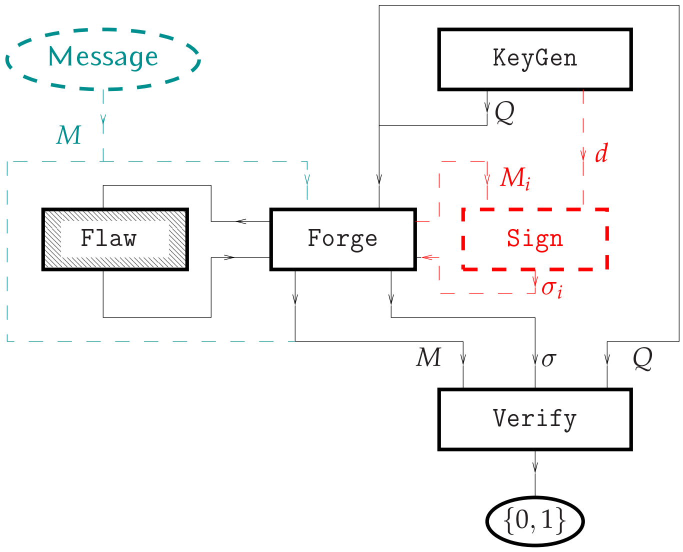
Summary of Necessary Security Conditions for (EC)DSA

1. **PRNG:**
   No (arithmetic) bias (ASF)
2. **Hash Function:**
   Collision-Resistant (AEF)
   2nd-Preimage-Resistant (ASF)
   One-Way (Preimage-Resistant) (PEF)
   Zero-Finder-Resistant (PEF)
   Rarely-Zero (PSF)
3. **Conversion Function:**
   Almost-Bijective (No large preimages) (PSF)
4. **Group:**
   Discrete logs hard (PSF)
   Semilogs hard (PSF)
5. **r-value:**
   Check for nonzeroness (PSF)

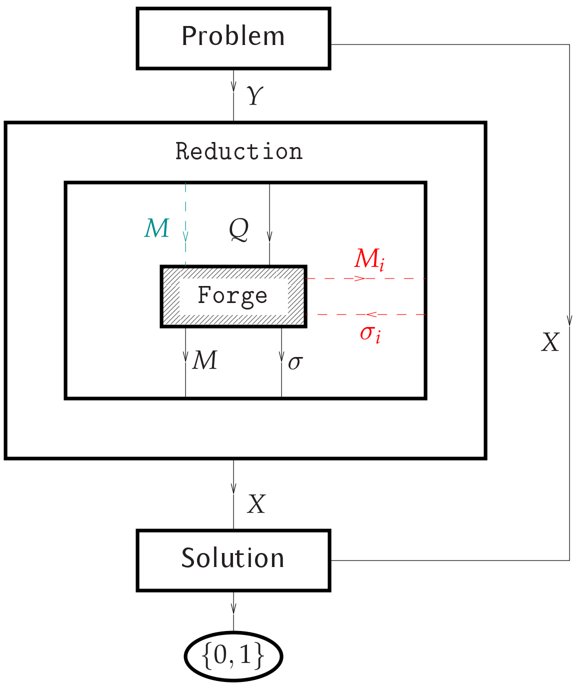
Quick Overview of Provable Security

1. **Flaw** $\Longrightarrow$ **Forgery**
2. **Forgery** $\Longrightarrow$ **Flaw**

Underlying Flaw $\Longrightarrow$ Forgery



Forgery $\Longrightarrow$ Underlying Flaw



Further Conditions Used Security Proofs

- One-time private key k:
  - Pseudorandom $k \xleftarrow{\$} \mathbb{Z}_n \setminus \{0\}$
- Hash function h:
  - "Smooth" hash
  - Ideal Hash h
- Group $\langle G \rangle$ and conversion f:
  - Almost Invertible f
  - Adaptive Semilogarithm Resistant
  - Ideal Group $\langle G \rangle$

Smooth[†] Hash

- There exists a distribution $\mathcal{U}$ of messages where
  - The distribution of $h(M)$ for $M \xleftarrow{\$} \mathcal{U}$, is indistinguishable from random elements in the range[‡] of h.
  - Given $e = h(M)$ where $M \xleftarrow{\$} \mathcal{U}$ it is information-theoretically infeasible to guess M because $h^{-1}(e) \cap \mathcal{U}$ is too large.
- If $\mathcal{U} = \{M \in \{0,1\}^{1000} : \text{Sha1}(M) < n\}$ then very plausible for h.
- Smooth + Collision-resistant $\implies$ One-way[§]

[†] Similar to notion of Luby, Cramer, Shoup

[‡] For $h = (\overline{\text{Sha1}} \bmod n)$:
If $n > 2^{160}$ then range is $[0, 2^{160} - 1] \subset \mathbb{Z}_n$.
If $n < 2^{160}$ then range is $[0, n-1] = \mathbb{Z}_n$.

[§] Stinson, Damgård

§5 Sufficient-in-part Conditions

Ideal Hash Function

- Rather than some concrete hash function, such as Sha1, consider a "random oracle" hash function[*]
- Conclusions DO NOT APPLY directly to instantiation with Sha1.
- Conclusions OFFER LIMITED[†] ASSURANCE that design does not use hash functions too poorly.

[*] Goldreich, Goldwasser, Micali ; Fiat, Shamir ; Bellare, Rogaway

[†] Canetti, Halevi, Goldreich

§5 Sufficient-in-part Conditions

Almost Invertible Conversion Function

- An **almost-inverse** g is a probabilistic algorithm with:
  - $g : \mathbb{Z}_n \xrightarrow{\$} \langle G \rangle \cup \{\texttt{fail}\}$
  - $f(g(z)) = z$ if $g(z) \neq \texttt{fail}$.
  - $\Pr\{g(z) \neq \texttt{fail} : z \xleftarrow{\$} \mathbb{Z}_n\} \geq 1/11$.
  - Distribution of $g(z)$ such that $g(z) \neq \texttt{fail}$ and $z \xleftarrow{\$} \mathbb{Z}_n$ is indistinguishable from uniform distribution of $\langle G \rangle$.
- Almost-invertible $\implies$ almost-bijective.
- Public key recovery: $Q = r^{-1}\left(sf^{-1}(r) - h(M)G\right)$.
- ECDSA: f is almost-invertible: solve a quadratic.
- DSA: f might be one-way, and thus not almost-invertible.

§5 Sufficient-in-part Conditions

Adaptive Semilogarithm Resistance

Adaptive Semilogarithm Problem:

- INPUT: P and G
- OUTPUT: A semilogarithm (t, u) of P to the base G
- HELP ALLOWED: A **supersigning** oracle:
  - INPUT: e_i such that $e_i \neq 0, 1$
  - OUPTUT: (t_i, u_i) such that $t_i = f\left(u_i^{-1}(e_i G + t_i P)\right)$.

Is the Adaptive Semilogarithm Problem HARD?
We will assume it is.

Ideal Group

- Rather than some concrete group isomorphic to $\mathbb{Z}_n$, such as an elliptic curve group, consider a "generic group"*

- Analogue to ideal hash.

- Representation of group elements controlled by a "random oracle"

- Ideal hash $\overset{?}{\not\Leftrightarrow}$ ideal group†

- Conclusions LIMITED analogously to ideal hash

*Shoup ; Jakobsson, Schnorr
†Anonymous referee

Security Proofs: Five Reductions

	Type of Security	Sufficient Conditions			
		h	f	$\langle G \rangle$	k
1	Pas+Exi	ZFR+OW	AI	GEN	PR
2	Act+Exi	ZFR+CR+S	AI	GEN	PR
3	Pas+Sel	RZ	SLR		PR
4	Act+Sel	ZFR+WCR	ASLR		PR
5	Act+Exi	ROM	SLR		PR

ZFR	Zero-finder resistant
RZ	Rarely zero
WCR	Weakly collision resistant (2nd preimages)
OW	One-way
CR	Collision Resistant
S	Smooth
ROM	Random oracle hash
SLR	Semilogarithm Resistant
ASLR	Adaptive Semilogarithm Resistant
AI	Almost Invertible
GEN	Generic group
PR	Pseudorandom

Reduction 1

Hash: Zero-Resistant, One-Way; **Group:** Ideal; **Conversion:** Almost-invertible
$\Longrightarrow$ Passive Existential Unforgeability.

Goal: Invert h at $e \overset{\$}{\leftarrow} \mathbb{Z}_n$:

- Run Passive Existential Forger with Ideal (Generic) Group

- A special group oracle response is **guessed** and depends on $f^{-1}(e)$.

- If guess is good, forged message is pre-image of e.

- Chance guess is good is $1/(\#$ of queries$)$.

Reduction 2

Hash: Zero-Resistant, Collision-Resistant, Smooth; **Group:** Generic; **Conversion:** Almost-invertible
$\Longrightarrow$ Active Existential Unforgeability.

Goal: Collision in h:

- Run Active Existential Forger with Ideal (Generic) Group

- Each group oracle response determined depends on $f^{-1}(h(\widetilde{M_i}))$ where $\widetilde{M_i} \overset{\$}{\leftarrow} \mathcal{U}$.

- Each signature oracle possible by reduction's knowledge of private key.

- Forged message M collides with some $\widetilde{M_i}$

Reduction 3

Hash: Rarely-zero; **Group and Conversion:** Semilog-resistant
$\Longrightarrow$ Passive Selective Unforgeability.

Goal: Semilog of $P \xleftarrow{\$} \langle G \rangle$:

- Run Passive Selective Forger on $M \xleftarrow{\$} \mathcal{S}$ and $Q \leftarrow h(M)P$.
- Convert forgery (r, s) under Q into semilogarithm $(r, h(M)^{-1}s)$ of P.

Reduction 4

Hash: Zero-Resistant and 2nd-Preimage-Resistant; **Group and Conversion:** Adaptive-Semilogarithm-Resistant
$\Longrightarrow$ Active Selective Unforgeability.

Goal: Semilog of $P \xleftarrow{\$} \langle G \rangle$ using **supersigning** oracle OR 2nd-Preimage of $M \xleftarrow{\$} \mathcal{S}$:

- Run Active Selective Forger on M and $Q \leftarrow h(M)P$.
- Answer signature query on M_i by querying the **supersigning** oracle on $h(M)^{-1}h(M_i)$.
- Convert forgery (r, s) under Q into semilogarithm $(r, h(M)^{-1}s)$ of P.

Reduction 5

Hash: Ideal; **Group and Conversion:** Semilog-Resistant
$\Longrightarrow$ Active Existential Unforgeability.

Goal: Semilog of $P \xleftarrow{\$} \langle G \rangle$:

- Pre-select $e \xleftarrow{\$} \mathbb{Z}_n$.
- Run active existential forger with public key $Q \leftarrow eP$.
- A special hash oracle response is **guessed** and given value e.
- If guess is good forgery convertible to semilogarithm as in Reductions 3 and 4.

So ...

1. **Summary**
2. **Remarks**
3. **Conclusion**

Summary

Security		h	f	$\langle G\rangle$	k
Pas+Sel	$\Rightarrow$	RZ	SLR		—
	$\Leftarrow$	RZ	SLR		—
Pas+Exi	$\Rightarrow$	ZFR+OW	SLR		—
	$\Leftarrow$	ZFR+OW	AI	GEN	—
Act+Sel	$\Rightarrow$	ZFR+WCR	SLR		NAB
	$\Leftarrow$	ZFR+WCR	ASLR		PR
Act+Exi	$\Rightarrow$	ZFR+CR	SLR		NAB
	$\Leftarrow$	ROM	SLR		PR
	$\Leftarrow$	ZFR+CR+S	AI	GEN	PR

Remarks

Some effects of changing $e \leftarrow h(M)$ to $e \leftarrow h(r||M)$ according to these and other provable security results are:

- PROS: If hash* or group† is modelled as ideal, then the provable security is better.

- CONS: If hash and group are not modelled as ideal, then it is more difficult to **separate** properties of hash and group.

If one isn't concerned about existential forgery, should one choose $n \approx |\mathcal{H}|^2$? For example, should one use a 384-bit elliptic curve with Sha1 to get 160-bit resistance against selective forgery (but 80-bit against existential)?

*Stern, Pointcheval; Schnorr, Jakobsson
†Schnorr, Jakobsson; Johnson, B.

Conclusion

- There are some minor differences between ECDSA and DSA in what conditions are necessary or sufficient for security.

- The security analysis is still incomplete because there is still a gap between the **sufficient** conditions and **necessary** conditions.