Limits in the Provable Security of ECDSA Signatures

Dominik Hartmann, Eike Kiltz

Horst Görtz Institute for IT Security, Ruhr University Bochum, Germany

Motivation

- (EC)DSA signatures is a standardized signature scheme and used *everywhere*
  - SSL/TLS
  - Blockchains (Bitcoin, Ethereum, . . .)
  - JSON Web Tokens (JWT)
  - . . .

Motivation

- (EC)DSA signatures is a standardized signature scheme and used *everywhere*
  - SSL/TLS
  - Blockchains (Bitcoin, Ethereum, . . .)
  - JSON Web Tokens (JWT)
  - . . .
- Comparatively few security results
- Existing results require strong idealization

GenDSA [FKP16]

Let $\mathcal{G} = (\mathbb{G}, p, g)$ be a group

GenDSA [FKP16]

Let $\mathcal{G} = (\mathbb{G}, p, g)$ be a group

<u>**Gen:**</u>

$x \xleftarrow{\$} \mathbb{Z}_p^*$; $X = g^x$
$\mathsf{vk} = X; \mathsf{sk} = x$
return $(\mathsf{vk}, \mathsf{sk})$

GenDSA [FKP16]

Let $\mathcal{G} = (\mathbb{G}, p, g)$ be a group

Gen:

$x \xleftarrow{\$} \mathbb{Z}_p^*$; $X = g^x$
vk $= X$; sk $= x$
return (vk, sk)

Sign(sk $= x, m$):

$r \xleftarrow{\$} \mathbb{Z}_p^*$; $R = g^r$
$h = H(m)$;

GenDSA [FKP16]

Let $\mathcal{G} = (\mathbb{G}, p, g)$ be a group and $f : \mathbb{G} \to \mathbb{Z}_p$ a "conversion function".

<u>Gen:</u>

$x \xleftarrow{\$} \mathbb{Z}_p^*;\ X = g^x$
$\mathsf{vk} = X; \mathsf{sk} = x$
return $(\mathsf{vk}, \mathsf{sk})$

<u>$\mathsf{Sign}(\mathsf{sk} = x, m)$:</u>

$r \xleftarrow{\$} \mathbb{Z}_p^*; R = g^r$
$h = H(m); t = f(R)$

GenDSA [FKP16]

Let $\mathcal{G} = (\mathbb{G}, p, g)$ be a group and $f : \mathbb{G} \to \mathbb{Z}_p$ a "conversion function".

<u>Gen:</u>

$x \xleftarrow{\$} \mathbb{Z}_p^*$; $X = g^x$
$\mathsf{vk} = X$; $\mathsf{sk} = x$
return $(\mathsf{vk}, \mathsf{sk})$

<u>$\mathsf{Sign}(\mathsf{sk} = x, m)$:</u>

$r \xleftarrow{\$} \mathbb{Z}_p^*$; $R = g^r$
$h = H(m)$; $t = f(R)$
$s = \frac{h + x \cdot t}{r}$

GenDSA [FKP16]

Let $\mathcal{G} = (\mathbb{G}, p, g)$ be a group and $f : \mathbb{G} \to \mathbb{Z}_p$ a "conversion function".

<u>**Gen:**</u>

$x \xleftarrow{\$} \mathbb{Z}_p^*$; $X = g^x$
$\mathsf{vk} = X$; $\mathsf{sk} = x$
return $(\mathsf{vk}, \mathsf{sk})$

<u>**Sign($\mathsf{sk} = x, m$):**</u>

$r \xleftarrow{\$} \mathbb{Z}_p^*$; $R = g^r$
$h = H(m)$; $t = f(R)$
$s = \frac{h + x \cdot t}{r}$
if $(t \stackrel{?}{=} 0) \vee (s \stackrel{?}{=} 0)$ **then**
  return $\perp$
return (s, t)

GenDSA [FKP16]

Let $\mathcal{G} = (\mathbb{G}, p, g)$ be a group and $f : \mathbb{G} \to \mathbb{Z}_p$ a "conversion function".

<u>**Gen:**</u>

$x \xleftarrow{\$} \mathbb{Z}_p^*$; $X = g^x$
$\mathsf{vk} = X$; $\mathsf{sk} = x$
return $(\mathsf{vk}, \mathsf{sk})$

<u>Sign$(\mathsf{sk} = x, m)$**:**</u>

$r \xleftarrow{\$} \mathbb{Z}_p^*$; $R = g^r$
$h = H(m)$; $t = f(R)$
$s = \frac{h + x \cdot t}{r}$
if $(t \stackrel{?}{=} 0) \vee (s \stackrel{?}{=} 0)$ **then**
  return $\perp$
return (s, t)

<u>Ver$(\mathsf{vk} = X, m, \sigma = (s, t))$**:**</u>

if $(t \stackrel{?}{=} 0) \vee (s \stackrel{?}{=} 0)$ **then**
  return $\perp$
$h = H(m)$
$t' = f\left(\left(g^h X^t\right)^{\frac{1}{s}}\right)$
return $t \stackrel{?}{=} t'$

The Conversion Function

- Conversion function is integral to security of (EC)DSA...
- ...yet very simple in practice:

The Conversion Function

- Conversion function is integral to security of (EC)DSA...
- ...yet very simple in practice:
  - DSA: Interprets bit representation of group element as integer (mod p)
  - ECDSA: Interprets bit representation of x-coordinate of curve point as integer (mod p)

The Conversion Function

- Conversion function is integral to security of (EC)DSA...
- ...yet very simple in practice:
  - DSA: Interprets bit representation of group element as integer (mod p)
  - ECDSA: Interprets bit representation of x-coordinate of curve point as integer (mod p)
- Completely breaks algebraic meaning
- Has no “unpredictability”

The Conversion Function

- Conversion function is integral to security of (EC)DSA...
- ...yet very simple in practice:
  - DSA: Interprets bit representation of group element as integer (mod p)
  - ECDSA: Interprets bit representation of x-coordinate of curve point as integer (mod p)
- Completely breaks algebraic meaning
- Has no "unpredictability"

Where is the problem?

The Problem

- Paillier & Vergnaud [PV05]: No security proof in standard model$^{(*)}$

The Problem

- Paillier & Vergnaud [PV05]: No security proof in standard model$^{(*)}$
- Three parts to idealize:

The Problem

- Paillier & Vergnaud [PV05]: No security proof in standard model$^{(*)}$
- Three parts to idealize:
  - hash function H as RO [FKP17]
  - group $\mathcal{G}$ as generic group [GS21]
  - conversion function f with programmable BRO [FKP16]

The Problem

- Paillier & Vergnaud [PV05]: No security proof in standard model$^{(*)}$
- Three parts to idealize:
  - hash function H as RO [FKP17]
  - group $\mathcal{G}$ as generic group [GS21]
  - conversion function f with programmable BRO [FKP16]

Modeling the conversion function [FKP16]

- $f : \mathbb{G} \to \mathbb{Z}_p$

$$\mathbb{G} \xrightarrow{f} \mathbb{Z}_p$$

Modeling the conversion function [FKP16]

- $f : \mathbb{G} \to \mathbb{Z}_p, \qquad f = \psi \circ \Pi \circ \varphi$

$$\begin{tikzcd}
\mathbb{G} \arrow[r, "\varphi"] \arrow[rrr, bend left, "f"] & \{0,1\}^L \arrow[r, "\Pi"] & [0{:}2^L - 1] \arrow[r, "\psi"] & \mathbb{Z}_p
\end{tikzcd}$$

Modeling the conversion function [FKP16]

- $f : \mathbb{G} \to \mathbb{Z}_p, \qquad f = \psi \circ \Pi \circ \varphi$

$$\begin{tikzcd} \mathbb{G} \arrow[r, "\varphi"] \arrow[rrr, bend left, "f"] & \{0,1\}^L \arrow[r, "\Pi"] & [0{:}2^L - 1] \arrow[r, "\psi"] & \mathbb{Z}_p \end{tikzcd}$$

- φ: 2-to-1 function
- Π: bijection
- ψ: arbitrary

Modeling the conversion function [FKP16]

- $f : \mathbb{G} \to \mathbb{Z}_p, \qquad f = \psi \circ \Pi \circ \varphi$

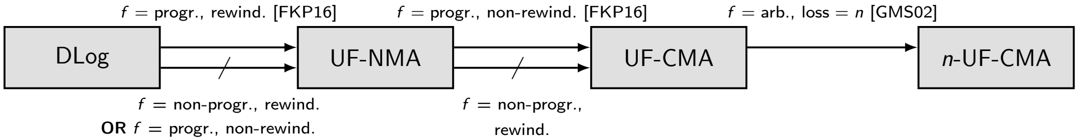
$$
\begin{tikzcd}
\mathbb{G} \arrow[r, "\varphi"] \arrow[rrr, bend left, "f"] & \{0,1\}^L \arrow[r, "\Pi"] & [0{:}2^L - 1] \arrow[r, "\psi"] & \mathbb{Z}_p
\end{tikzcd}
$$

- φ: 2-to-1 function
- Π: bijection $\leftarrow$ modeled as bijective random oracle
- ψ: arbitrary

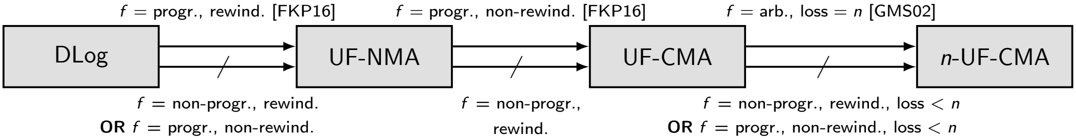
Results

$$\text{DLog} \xrightarrow{f\,=\,\text{progr., rewind. [FKP16]}} \text{UF-NMA} \xrightarrow{f\,=\,\text{progr., non-rewind. [FKP16]}} \text{UF-CMA} \xrightarrow{f\,=\,\text{arb., loss}\,=\,n\ \text{[GMS02]}} n\text{-UF-CMA}$$

Results


f = progr., rewind. [FKP16]
f = progr., non-rewind. [FKP16]
f = arb., loss = n [GMS02]
DLog
UF-NMA
UF-CMA
n-UF-CMA
f = non-progr., rewind.
OR f = progr., non-rewind.
f = non-progr.,
rewind.

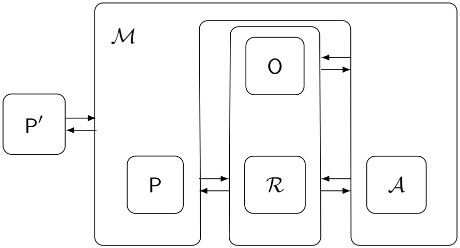
Results



f = progr., rewind. [FKP16]
f = progr., non-rewind. [FKP16]
f = arb., loss = n [GMS02]
DLog
UF-NMA
UF-CMA
n-UF-CMA
f = non-progr., rewind.
OR f = progr., non-rewind.
f = non-progr.,
rewind.
f = non-progr., rewind., loss < n
OR f = progr., non-rewind., loss < n

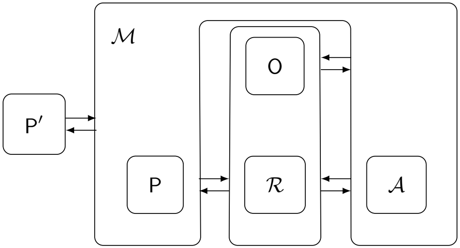
Meta Reduction

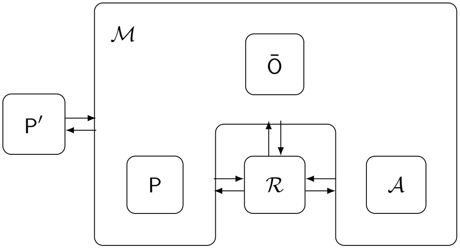
- "Reduction against the reduction"



Meta Reduction

- "Reduction against the reduction"





Simulating an Adversary

- Problem 1: Simulate successfull adversary without secret key

Simulating an Adversary

- Problem 1: Simulate successfull adversary without secret key
- Solution: Use the assumption attacked by the meta-reduction
  - Free-Base One-More Discrete Logarithm assumption (FBOMDL)
  - Provides access to a DLog oracle (relative to a chosen base element)
  - only usable if we get more challenges

Simulating an Adversary

- Problem 1: Simulate successfull adversary without secret key
- Solution: Use the assumption attacked by the meta-reduction
  - Free-Base One-More Discrete Logarithm assumption (FBOMDL)
  - Provides access to a DLog oracle (relative to a chosen base element)
  - only usable if we get more challenges
- Problem 2: How to extract all solutions?

Simulating an Adversary

- Problem 1: Simulate successfull adversary without secret key
- Solution: Use the assumption attacked by the meta-reduction
  - Free-Base One-More Discrete Logarithm assumption (FBOMDL)
  - Provides access to a DLog oracle (relative to a chosen base element)
  - only usable if we get more challenges
- Problem 2: How to extract all solutions?
- Solution: AGM and clever simulation of $(\bar{\Pi}, \bar{\Pi}^{-1})$

Interpretation

- Is ECDSA now broken?

Interpretation

- Is ECDSA now broken?
  - No, but the proofs require strong, potentially unrealistic assumptions

Interpretation

- Is ECDSA now broken?
  - No, but the proofs require strong, potentially unrealistic assumptions
- Isn't the q-FBOMDL assumption really strong?

Interpretation

- Is ECDSA now broken?
  - No, but the proofs require strong, potentially unrealistic assumptions
- Isn't the q-FBOMDL assumption really strong?
  - Yes, but only used for meta-reduction

Interpretation

- Is ECDSA now broken?
  - No, but the proofs require strong, potentially unrealistic assumptions
- Isn't the q-FBOMDL assumption really strong?
  - Yes, but only used for meta-reduction
- Can we get around these problems?

Interpretation

- Is ECDSA now broken?
  - No, but the proofs require strong, potentially unrealistic assumptions
- Isn't the q-FBOMDL assumption really strong?
  - Yes, but only used for meta-reduction
- Can we get around these problems? Yes
  - Find non-algebraic/non-black-box reductions

Interpretation

- Is ECDSA now broken?
  - No, but the proofs require strong, potentially unrealistic assumptions
- Isn't the q-FBOMDL assumption really strong?
  - Yes, but only used for meta-reduction
- Can we get around these problems? Yes
  - Find non-algebraic/non-black-box reductions
  - Use stronger assumptions

Thank you!

Eprint: ia.cr/2023/914