Modifications of ECDSA

John Malone-Lee and Nigel P. Smart

University of Bristol, Department of Computer Science,
Merchant Venturers Building,
Woodland Road,
Bristol, BS8 1UB, UK.
{malone, nigel}@cs.bris.ac.uk

Abstract. We describe two variants of ECDSA one of which is secure, in the random oracle model, against existential forgery but suffers from the notion of duplicate signatures. The second variant is also secure against existential forgery but we argue that it is likely to possess only four natural duplicate signatures. Our variants of ECDSA are analogous to the variants of DSA as proposed by Brickell *et al.* However, we show that the ECDSA variants have better exact security properties.

1 Introduction

In 1984, Goldwasser, Micali and Rivest [4], [5] introduced the notion of *existential forgery against adaptive chosen-message attack* for public key signature schemes. This notion has now become the *de facto* security definition for digital signature algorithms, against which all new schemes are measured. The definition involves a game in which the adversary is given a target user's public key and is asked to produce a valid signature, on any message, with respect to this public key. The adversary is given access to an oracle which will produce signatures on messages of the adversary's choice, in which case the signature output by the adversary at the end should clearly not have resulted from a query to its oracle.

Our work is motivated by the wish to create tighter security reductions for modified forms of ECDSA. This builds on the earlier work of Brickell *et al.* [2] who looked at the security of DSA and various minor modifications thereof. The work of Brickell *et al.* itself builds upon the earlier work of Pointcheval and Stern [7].

We shall describe a minor modification of ECDSA (which we call ECDSA-II), similar to the DSA-II variant of Brickell *et al.*, which is secure against existential forgery in the random oracle model. Our main contribution is that the tightness of our security reduction for ECDSA-II is better than that obtainable for DSA-II. However, ECDSA-II (just as ECDSA) suffers from the notion of duplicate signatures, as introduced in [8]. Duplicate signatures should not necessarily be considered a security weakness but they point out possible problems with the underlying design of the signature scheme.

The purpose of presenting ECDSA-II is to demonstrate the difference in the security result with DSA-II and to show why our final modification is better from

K. Nyberg and H. Heys (Eds.): SAC 2002, LNCS 2595, pp. 1–12, 2003.

the point of view of existential forgery. Finally we present ECDSA-III, which we shall show is secure against existential forgery in the random oracle model, with a tighter result than one could obtain for ECDSA-II. In addition we shall argue that ECDSA-III does not suffer from duplicate signatures. In fact by trying to remove the possibility of duplicate signatures we obtain a tighter security reduction. This demonstrates that removing anomalies in signature algorithms may lead to better provable security results, even when the anomalies are not security weaknesses.

The paper is structured as follows: we first define ECDSA and then we present DSA-II and ECDSA-II. Both schemes can be proved secure in the random oracle model against active adversaries. The proof technique requires *The Improved Forking Lemma* [2], and relies in the case of DSA-II on a heuristic assumption as to the distribution of the conversion function. No such heuristic is required for ECDSA-II and the resulting security reduction is also tighter than that for DSA-II. Subsequently we define ECDSA-III, which is another minor modification. We show that this is also secure, with an even tighter security reduction. Finally in the Appendix we give a uniform analogue of *The Improved Forking Lemma* that we call *The Uniform Multiple Forking Lemma*.

2 Definition of ECDSA

To use ECDSA, as defined in the ANSI [1] and other standards, one first picks an elliptic curve E over a finite field $\mathbb{F}_q$ whose order is equal to a prime n times a small cofactor c, i.e.

$$\#E(\mathbb{F}_q) = c \cdot n.$$

In addition, a base point $P \in E(\mathbb{F}_q)$ is chosen of order n. Note that, while users are free to choose their own individual base points, it is argued in [6] that they should be set by some central authority (alternatively one could generate the base point verifiably at random).

Each user has a private key $x \in \{1, \ldots, n-1\}$ and a public key

$$Z = xP.$$

ECDSA uses a hash function $H : \{0,1\}^* \to \{1, \ldots, n-1\}$. The algorithm itself is then given by:

Sign	**Verify**
1. $k \leftarrow \{1, \ldots, n-1\}$	1. $h \leftarrow H(m)$
2. $Q \leftarrow kP$	2. $a \leftarrow h/s \pmod n$
3. $r \leftarrow \text{xcoord}(Q) \pmod n$	3. $b \leftarrow r/s \pmod n$
4. $h \leftarrow H(m)$	4. $Q \leftarrow aP + bZ$
5. $s \leftarrow (h + x \cdot r)/k \pmod n$	5. $t \leftarrow \text{xcoord}(Q) \pmod n$
6. Output (r, s)	6. Accept iff $r = t$

A duplicate signature, see [8], for ECDSA is a pair of messages (m_1, m_2) and a signature (r, s) such that

$$H(m_1) \neq H(m_2)$$

and such that the pair (r, s) is a valid signature on both messages.

3 ECDSA - II

In [2] a modification of DSA is given, called DSA-II, which replaces the hash function evaluation $h = H(m)$ with $h = H(m\|r)$ where

$$r = \left(g^k \pmod p\right) \pmod q.$$

The authors of [2] claim that the map

$$k \to \left(g^k \pmod p\right) \pmod q$$

is likely to be $(\log q)$-collision free, in that it is impossible to find $\log q$ different values of $k \in \{1, \ldots, q-1\}$ which map to the same number under the above map. Using this heuristic, and in the random oracle model, the authors of [2] prove the result below.

Theorem 1. *Suppose an adversary A against DSA-II exists which succeeds with probability $\epsilon > 4/q$ after Q queries to the random oracle H, then one can solve the discrete logarithm problem modulo p using fewer than*

$$25Q(\log q)(\log(2\log q))/\epsilon$$

replays of A with probability greater than $1/100$.

The proof uses a generalisation of *The Forking Lemma* from [7], *The Improved Forking Lemma* in [2], but it requires $\log q$ signatures on the same message with different random oracles to be produced.

The Improved Forking Lemma applies to *Trusted El Gamal Type Signature Schemes*, as defined in [2]. Here we define the analogous notion for schemes based on elliptic curves rather than finite fields: *Elliptic Curve Trusted El Gamal Type Signature Schemes* (ECTEGTSS).

Definition 1 (ECTEGTSS). A signature scheme is an ECTEGTSS if it has the following properties:

- The underlying group is from an elliptic curve E over a finite field $\mathbb{F}_q$ whose order is equal to a prime n times a small cofactor c, i.e. $\#E(\mathbb{F}_q) = c \cdot n$. A base point $P \in E(\mathbb{F}_q)$ of order n is given.
- It uses two functions G and H, with ranges $\mathcal{G}$ and $\mathcal{H}$ respectively. For security analysis the function H is modelled as a random oracle and G requires some practical properties such as (multi)-collision-resistance or (multi)-collision-freeness.
- There are three functions:

  $$F_1(\mathbb{Z}_n, \mathbb{Z}_n, \mathcal{G}, \mathcal{H}) \to \mathbb{Z}_n,\ \mathbb{F}_2(\mathbb{Z}_n, \mathcal{G}, \mathcal{H}) \to \mathbb{Z}_n,\ F_3 : (\mathbb{Z}_n, \mathcal{G}, \mathcal{H}) \to \mathbb{Z}_n$$

  satisfying for all $(k, x, r, h) \in (\mathbb{Z}_n, \mathbb{Z}_n, \mathcal{G}, \mathcal{H})$,

  $$\mathbb{F}_2(F_1(k, x, r, h), r, h) + x \cdot F_3(F_1(k, x, r, h), r, h) = k \bmod n.$$

- Each user has private and public keys x, Z such that $Z = xP$.
- To sign a message m, the signer Alice picks k at random from $\mathbb{Z}_n^*$, computes $Q = kP$ and $r = G(Q)$. She then gets $h = H(m||r)$ and computes $s = F_1(k, x, r, h)$. The signature on m is (s, r, h), although (s, r) is enough in practice since h may be recovered from m and r.
- To verify the signature (s, r, h) on a message m the verifier Bob computes $e_P = F_2(s, r, h)$, $e_Z = F_3(s, r, h)$ and finally $W = e_P P + e_Z Z$. He then checks that $r = G(W)$ and $h = H(m||r)$.
- The functions F_2 and F_3 must satisfy the following one-to-one condition: for given r, e_P and e_Z, there exists a unique pair (h, s) such that

$$e_P = F_2(s, r, h) \text{ and } e_Z = F_3(s, r, h).$$

  Furthermore, this pair is easy to find.

■

It is easily verified that the proof of *The Improved Forking Lemma* in [2] applies to ECTEGTSSs.

The analogue to ECDSA of DSA-II is the following signature algorithm, which we call ECDSA-II.

Sign	**Verify**
1. $k \leftarrow \{1, \ldots, n-1\}$	1. $h \leftarrow H(m\|r)$
2. $Q \leftarrow kP$	2. $a \leftarrow h/s \pmod n$
3. $r \leftarrow \text{xcoord}(Q) \pmod n$	3. $b \leftarrow r/s \pmod n$
4. $h \leftarrow H(m\|r)$	4. $Q \leftarrow aP + bZ$
5. $s \leftarrow (h + x \cdot r)/k \pmod n$	5. $t \leftarrow \text{xcoord}(Q) \pmod n$
6. Output (r, s)	6. Accept iff $r = t$

It is easily verified that ECDSA-II is an ECTEGTSS with:

$$F_1(k, x, r, h) = (h + x \cdot r)/k \bmod n = s$$
$$F_2(s, r, h) = h/s \bmod n$$
$$F_3(s, r, h) = r/s \bmod n$$

where $Q = kP$, $r = \text{xcoord}(Q) \bmod n$ and $h = H(m||r)$.

The scheme still exhibits duplicate signatures because the function

$$G : k \rightarrow \text{xcoord}(kP) \pmod n$$

possesses trivial collisions, in that k and $-k$ always map to the same point. However, we also have that if

$$y = \lfloor q/n \rfloor + 1$$

and if

$$G(k_1) = G(k_2) = \ldots = G(k_y)$$

then there must exist $i, j \in \{1, \ldots, y\}$ with $i \neq j$ such that

$$k_i = \pm k_j.$$

Since we have

$$c \cdot n = \#E(\mathbb{F}_q) \geq q + 1 - 2\sqrt{q}$$

and for most elliptic curves used in "real life" we have $c \leq 4$, we deduce

$$y = \lfloor q/n \rfloor + 1 \leq \frac{4q}{q + 1 - 2\sqrt{q}} + 1 \leq 6.$$

This leads us to the following result:

Theorem 2. *Suppose an adversary A against ECDSA-II exists which succeeds with probability $\epsilon > 4/q$ after Q queries to the random oracle H, then one can solve the discrete logarithm problem in $E(\mathbb{F}_q)$ using fewer than*

$$150Q \log 12/\epsilon$$

replays of A with probability greater than $1/100$.

Proof. As in Theorem 1 we apply *The Improved Forking Lemma* from [2] to obtain 6 valid signatures on the same message m, each with a different random oracle. Denote these signatures

$$(r_i, h_i, s_i)$$

where

$$\begin{aligned} r_i &= \text{xcoord}(k_i P) \pmod{n}, \\ h_i &= H_i(m \| r_i), \\ s_i &= (h_i + x \cdot r_i)/k_i \pmod{n}. \end{aligned}$$

We have for all these signatures that $r = r_i = r_j$, and so there exists two indices i and j, with $i \neq j$, such that

$$k_i = \pm k_j.$$

Then using the equality

$$(h_i + x \cdot r)/s_i = \pm (h_j + x \cdot r)/s_j \pmod{n}$$

we obtain two possibilities for the discrete logarithm x of the public key. The correct value of the discrete logarithm may then be determined using one point multiplication.

Notice how this only requires 6 different signatures as opposed to the $\log q$ different signatures in the result for DSA-II. In addition notice that it is the absence of collision resistance in G which makes the above security reduction tighter.

The above result holds for passive adversaries, a similar result for active adversaries can be deduced by providing the obvious signing simulator.

4 ECDSA - III

We now present a version of ECDSA which we call ECDSA-III. We shall show that it does not exhibit general duplicate signatures. In addition the security reduction against existential forgeries is tighter for ECDSA-III than for ECDSA-II. The alteration is to replace

$$r \leftarrow \text{xcoord}(kP) \pmod{n}$$

by

$$r \leftarrow X + Y$$

where $Q = (X, Y) = kP$. Notice how r is now treated as an element of $\mathbb{F}_q$ and not $\mathbb{F}_n^*$ and how the value of r depends on both the x and y coordinates of the point kP.

The precise details of ECDSA-III we give below, which one should notice is only marginally less efficient in terms of bandwidth and CPU time than standard ECDSA.

Sign

1. $k \leftarrow \{1, \ldots, n-1\}$
2. $Q = (X, Y) \leftarrow kP$
3. $r \leftarrow X + Y$
4. $h \leftarrow H(m \| r)$
5. $s \leftarrow (h + x \cdot r)/k \pmod{n}$
6. Output (r, s)

Verify

1. $h \leftarrow H(m \| r)$
2. $a \leftarrow h/s \pmod{n}$
3. $b \leftarrow r/s \pmod{n}$
4. $Q = (X, Y) \leftarrow aP + bZ$
5. Accept iff $r = X + Y$

It is easily verified that ECDSA-III is an ECTEGTSS with:

$$F_1(k, x, r, h) = (h + x \cdot r)/k \bmod n = s$$
$$F_2(s, r, h) = h/s \bmod n$$
$$F_3(s, r, h) = r/s \bmod n$$

where $Q = kP = (X + Y)$, $r = X + Y$ and $h = H(m||r)$.

Notice that the equation $X + Y = t$ will intersect the curve in at most three points. This leads us to the following improved security reduction:

Theorem 3. *Suppose an adversary A against ECDSA-III exists which succeeds with probability $\epsilon > 4/q$ after Q queries to the random oracle H, then one can solve the discrete logarithm problem in $E(\mathbb{F}_q)$ using fewer than*

$$100Q \log 8/\epsilon$$

replays of A with probability greater than $1/100$.

Proof. Again we apply *The Improved Forking Lemma* from [2] to obtain four signatures with the same value of r. Two of these signatures correspond to points, $Q_1 = k_1 P$ and $Q_2 = k_2 P$, with $k_1 = \pm k_2$. We may now recover the discrete logarithm of the public key Z in the obvious way.

As usual one can simulate the signing oracles so as to obtain a similar result for active adversaries.

The way in which *The Improved Forking Lemma* uses the adversary A to produce multiple signatures on the same message depends on A's probability of success ϵ, and the number of oracle queries it makes Q. This makes the resulting reduction non-uniform. We give the following uniform reduction for ECDSA-III.

Theorem 4. *Suppose an adversary A against ECDSA-III exists which succeeds in time T and with probability $\epsilon > 14Q/q$ after Q queries to the random oracle H, then one can solve the discrete logarithm problem in $E(\mathbb{F}_q)$ using a probabilistic algorithm in expected time*

$$T' \leq 1984506 \cdot Q \cdot T/\epsilon.$$

Proof. We apply *The Uniform Multiple Forking Lemma* from the appendix with $y = 4$ and $2^k = q$ and reason as in the proof of Theorem 3.

Note that for schemes where we may usefully apply *The Uniform Multiple Forking Lemma*, the efficiency of the resulting security reduction depends very much on the parameter y, the number of signatures required on the same message. The smaller the value of y, the better the reduction. We require $\log q$ signatures for DSA-II, 6 signatures for ECDSA-II, and only 4 for ECDSA-III. Therefore, as in the non-uniform case, the uniform reduction given in Theorem 4 for ECDSA-III is the tightest among the reductions using this method for the three schemes considered in this paper.

We now turn to discussing whether ECDSA-III is resistant to duplicate signatures. We cannot give a security proof but give an informal argument.

We wish to show that it is hard to find two elliptic curve points $Q_1 = (x_1, y_1)$ and $Q_2 = (x_2, y_2)$ such that one knows the respective discrete logarithms $Q_i = k_i P$ and such that

$$x_1 + y_1 = x_2 + y_2.$$

Intuitively this can only happen when the line

$$L(t) : X + Y = t$$

for some constant t is geometrically related to the group law linking Q_1 and Q_2. If $L(t)$ is a tangent at Q_1 and we know the discrete logarithm k_1 then we know that $L(t)$ intersects the curve in one other point, say $Q_2 = k_2 P$, of the required form and that $k_2 = (-2k_1) \pmod{n}$. Hence we need to avoid points where $L(t)$ is a tangent. But for all possible values of t the line $L(t)$ is only a tangent for at most four points on any given elliptic curve.

In ECDSA all values of r could be members of a trivial duplicate signature, for ECDSA-III we see that only four possible values of r can be members of a trivial duplicate signature.

Now assume that we have a value of t such that $L(t)$ is not a tangent to the curve. Suppose it intersects the curve at Q_1 and that we know the discrete logarithm of Q_1 with respect to P. About fifty percent of the time there will be no other $\mathbb{F}_q$-point on the curve which lies on the line $L(t)$, in which case Q_1 cannot be part of a duplicate signature. For the other fifty percent of the time we obtain two other elliptic curve points Q_2 and Q_3. To use these points to obtain a duplicate signature it would appear we need to extract their discrete logarithm with respect to P. Although we cannot prove that this is the only way to obtain duplicate signatures it seems likely to be the case.

5 Conclusion

We have shown that a modified form of ECDSA has a tighter security reduction than a similarly modified form of DSA. In addition we have presented a second modified form of ECDSA which not only has an even tighter security reduction, it also does not suffer from the phenomenon of duplicate signatures.

6 Acknowledgements

Many thanks to David Pointcheval for helpful correspondence during this work.

References

1. ANSI X9.62. *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm* (ECDSA), 1999.
2. E. Brickell, D. Pointcheval, S. Vaudenay and M. Yung. Design validations for discrete logarithm based signature schemes. *Public Key Cryptography 2000*, Springer-Verlag LNCS 1751, 276–292, 2000.
3. D. Brown. Generic groups, collision resistance and ECDSA. Preprint, 2001.
4. S. Goldwasser, S. Micali and R. Rivest. A "paradoxical" solution to the signature problem. *Proc. 25th Symposium on Foundations of Computer Science*, 441–448, 1984.
5. S. Goldwasser, S. Micali and R. Rivest. A digital signature scheme secure against adaptive chosen ciphertext attacks. *SIAM J. Computing*, **17**, 28–308, 1988.
6. A. Menezes and N.P. Smart. Security of signature schemes in a multi-user setting. Preprint 2001.
7. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, **13**, 361–396, 2000.
8. D. Pointcheval, J. Stern, J. Malone-Lee and N.P. Smart. Flaws in Security Proofs. To appear *Advances in Cryptology - CRYPTO 2002*.

Appendix

In this Appendix we prove a generalisation of the forking Lemma of [7]. Our result applies to *generic signature schemes*, as defined below.

Definition 2. *[7] A* **generic signature scheme** *is a signature scheme (G, V, S) such that on input of the message m, the signing algorithm S produces a signature $(m, \sigma_1, h, \sigma_2)$, where σ_1 randomly takes its values in a large set, h is the hash value of $m||\sigma_1$, and σ_2 depends on σ_1, m, and h. Here and henceforth $||$ denotes concatenation.*

Theorem 5 (The Uniform Multiple Forking Lemma). *Let (G, S, V) be a generic digital signature scheme with security parameter k. Let A be a probabilistic polynomial time Turing machine whose input consists of public data and which can make $Q > 0$ queries to a random oracle $\mathcal{O}$. Assume that, within time bound T, the attacker A produces a valid signature $(m, \sigma_1, h, \sigma_2)$ with probability $\epsilon \geq xQ/2^k$. Then, for y such that $x - 1 > 4(y - 1)$, there is a machine M that by using A can produce y valid signatures,*

$$(m, \sigma_1, h, \sigma_2) \text{ and } (m, \sigma_1, h^i, \sigma_2^i)$$

for $i = 1, \ldots, y - 1$, with h, h^i all distinct. The expected running time of M is upper bounded by

$$21 \cdot \frac{x}{x-1} \cdot \sigma(x, y) \cdot \frac{(1 + \gamma(y))^3}{\gamma(y)(\gamma(y) - 1)^2} \cdot \frac{QT}{\epsilon},$$

where

$$\sigma(x, y) = \sum_{k=1}^{y-1} \frac{x}{x - 1 - 4k} \text{ and } \gamma(y) = 1 - \left(\frac{1}{4}\right)\left(\frac{3}{4}\right)\left(\frac{3}{5}\right)^{y-1}.$$

Our proof uses the following lemma from [7]:

Lemma 1 (The Splitting Lemma). *Let $A \subset X \times Y$ be such that $\Pr[A] \geq \epsilon$. Define*

$$B = \{(x, y) \in X \times Y : \Pr_{y' \in Y}[(x, y') \in A] \geq \epsilon/2\}.$$

We have the following:

1. $\forall (x, y) \in B, \Pr_{y' \in Y}[(x, y') \in A] \geq \epsilon/2$.
2. $\Pr[B|A] \geq 1/2$.

We now return to the proof of the main theorem.

Proof (Of Theorem 5). The attacker A is a probabilistic polynomial time Turing machine with random tape ω that mounts a no-message attack on (G, S, V). During the attack A makes a polynomial number of queries to the random oracle $\mathcal{O}$. Let us denote these queries $\mathcal{Q}_1, \ldots, \mathcal{Q}_Q$, and the corresponding responses $\rho_1, \ldots, \rho_Q$. We will assume that A stores the query and answer pairs in a table and so the queries are all distinct. Clearly a random choice of $\mathcal{O}$ corresponds to random choices for $\rho_1, \ldots, \rho_Q$.

For a random choice of $(\omega, \mathcal{O})$, A outputs a valid signature $(m, \sigma_1, h, \sigma_2)$ with probability ϵ. Since $\mathcal{O}$ is random, the probability for h to be equal to $\mathcal{O}(m||\sigma_1)$ is

less than $1/2^k$ unless A made the query $m||\sigma_1$ during its attack. We let $Ind(\omega, \mathcal{O})$ be the index of the query $m||\sigma_1$ and we let $Ind(\omega, \mathcal{O}) = \infty$ if this query is never made. We define the sets

$$S = \{(\omega, \mathcal{O}) : A^{\mathcal{O}}(\omega) \text{ succeeds and } Ind(\omega, \mathcal{O}) \neq \infty\} \text{ and}$$
$$S_i = \{(\omega, \mathcal{O}) : A^{\mathcal{O}}(\omega) \text{ succeeds and } Ind(\omega, \mathcal{O}) = i\} \text{ for } i \in \{1, \ldots, Q\}.$$

For index i let $\mathcal{O}|_i$ denote the restriction of $\mathcal{O}$ to queries of index strictly less than i, and let $\mathcal{O}|^i$ denote the restriction of $\mathcal{O}$ to queries of index greater or equal to i.

We define the parameter

$$\alpha(y) = \frac{1}{2\gamma(y)} + \frac{1}{2}. \tag{1}$$

The machine M is now described in figure 1.

Fig. 1. Machine M

algorithm M
1. $j = 1$
2. run A until, on input of a pair $(\omega, \mathcal{O}) \in S$, it outputs a forgery
   call such a forgery ''successful''
   denote the number of calls made to A to obtain a successful forgery by N_j, and denote $Ind(\omega, \mathcal{O})$ by β
3. for $k = 1, \ldots, y-1$:
   i. run A until it produces a new successful forgery, or at most $20N_j\alpha(y)^j\delta_k$ times, where $\delta_k = x/(x-1-4k)$
      for each run use the same ω as above and choose $\mathcal{O}^{j,k}$ randomly subject to $\mathcal{O}^{j,k}|_\beta = \mathcal{O}|_\beta$
   ii. if A has not produced a new successful forgery goto 4
      else increment k
4. if A has produced y successful forged signatures return these
   else increment j and goto 2

The sets in $\{S_i : i \in \{1, \ldots, Q\}\}$ form a partition of S. With our definitions we have

$$\nu = \Pr[S] \geq \epsilon - 1/2^k \geq \epsilon - \epsilon/xQ \geq \epsilon(x-1)/x. \tag{2}$$

Also

$$\Pr[N_j \geq 1/5\nu] = (1-\nu)^{\lceil 1/5\nu \rceil - 1} > 3/4. \tag{3}$$

Let I be the set of the most likely indices,

$$I = \{i : \Pr[S_i|S] \geq 1/2Q\}.$$

Define the sets

$$\Omega_i = \left\{(\omega, \mathcal{O}) : \Pr_{\mathcal{O}\clubsuit}\left[(\omega, \mathcal{O}|_{\mathrm{i}}, \mathcal{O}|^{\mathrm{i}'}) \in \mathrm{S_i}\right] \geq \nu/4\mathrm{Q}\right\}.$$

For $i \in I$ we have

$$\Pr[S_i] = \Pr[S_i \cap S] = \Pr[S_i|S] \cdot \Pr[S] \geq \nu/2Q,$$

and so by the Splitting Lemma $\Pr[\Omega_i|S_i] \geq 1/2$.

Since all the subsets S_i are disjoint,

$$\begin{aligned}
&\Pr_{\omega,\mathcal{O}}[(\exists \mathrm{i} \in \mathrm{I})\ (\omega, \mathcal{O}) \in \Omega_{\mathrm{i}} \cap \mathrm{S_i}|\mathrm{S}] \\
&= \Pr\ \Big[\bigcup_{\mathrm{i}\in\mathrm{I}} (\Omega_{\mathrm{i}} \cap \mathrm{S_i})|\mathrm{S}\Big] = \sum_{\mathrm{i}\in\mathrm{I}} \Pr[\Omega_{\mathrm{i}} \cap \mathrm{S_i}|\mathrm{S}] \\
&= \sum_{i\in I} \Pr[\Omega_i|S_i] \cdot \Pr[S_i|S] \geq \ \sum_{i\in I} \Pr[S_i|S]\Big)/2 \geq 1/4. \qquad (4)
\end{aligned}$$

Define $l = \lceil \log_{\alpha(y)} Q \rceil$. For any $j \geq l$ and any $1 \leq k \leq y$, whenever $N_j \geq 1/5\nu$ we have

$$\begin{aligned}
20N_j\alpha(y)^j\delta_k &\geq 20 \cdot \frac{1}{5\nu} \cdot \alpha(y)^{\lceil \log_{\alpha(y)} Q \rceil} \cdot \frac{x}{x-1-4k} \\
&> \frac{4xQ}{\epsilon(x-1-4k)}. \qquad (5)
\end{aligned}$$

At step 3 of M when $(\omega, \mathcal{O}) \in \Omega_\beta \cap S_\beta$ we have

$$\begin{aligned}
&\Pr_{\mathcal{O}^{\mathrm{j,k}}\clubsuit}[(\omega, \mathcal{O}^{\mathrm{j,k}}) \in \mathrm{S}_\beta \text{ and } \rho_\beta^{\mathrm{j,k}} \neq \rho_\beta, \rho_\beta^{\mathrm{j,k}} \neq \rho_\beta^{\mathrm{j},1}, \ldots, \rho_\beta^{\mathrm{j,k}} \neq \rho_\beta^{\mathrm{j,k-1}}] \\
&\geq \Pr_{\mathcal{O}^{\mathrm{j,k}}\clubsuit}[(\omega, \mathcal{O}^{\mathrm{j,k}}) \in \mathrm{S}_\beta] - \Pr_{\mathcal{O}^{\mathrm{j,k}}\clubsuit}[\rho_\beta^{\mathrm{j,k}} = \rho_\beta] - \sum_{\mathrm{i}=1}^{\mathrm{k}-1} \Pr_{\mathcal{O}^{\mathrm{j,k}}\clubsuit}[\rho_\beta^{\mathrm{j,k}} = \rho_\beta^{\mathrm{j,i}}] \\
&\geq \nu/4Q - k/2^k \geq \epsilon(x-1)/4xQ - \epsilon k/xQ = \epsilon(x-1-4k)/4xQ \qquad (6)
\end{aligned}$$

From (5) and (6) we know that, for $j \geq l$ and $N_j \geq 1/5\nu$, the probability of getting $y-1$ successful forks after at most $\sum_{k=1}^{y-1} 4xQ/\epsilon(x-1-4k)$ runs of A at step 3 is greater or equal to

$$\begin{aligned}
&\prod_{k=1}^{y-1} \left(1 - \left(1 - \frac{\epsilon(x-1-4k)}{4xQ}\right)^{4xQ/\epsilon(x-1-4k)}\right) \\
&\geq (1-e^{-1})^{y-1} > \left(\frac{3}{5}\right)^{y-1}. \qquad (7)
\end{aligned}$$

Recall from (4) that at step 2 of M when A produces a forgery using $(\omega, \mathcal{O}) \in S$ with $Ind(\omega, \mathcal{O}) = \beta$ then, with probability at least 1/4, $(\omega, \mathcal{O}) \in \Omega_\beta \cap S_\beta$.

Combining this fact with (3) and (7) we have that for any $t \geq l$ the probability for J to be greater or equal to t is less than

$$\left(1 - \left(\frac{1}{4}\right)\left(\frac{3}{4}\right)\left(\frac{3}{5}\right)^{y-1}\right)^{t-l} = \gamma(y)^{t-l}. \tag{8}$$

Let J denote the final value of j during an execution of M and let N be the total number of calls made to A. We want to compute an upper bound on the expectation on N. We have

$$\begin{aligned}\mathrm{E}[N|J=t] &\leq \sum_{j=1}^{t} \big(\mathrm{E}[N_j] + 20\mathrm{E}[N_j]\alpha(y)^j \sum_{k=1}^{y-1} \delta_k\big) \\ &= \sum_{j=1}^{t} \big(\mathrm{E}[N_j] + 20\mathrm{E}[N_j]\alpha(y)^j \sigma(x,y)\big),\end{aligned}$$

and,

$$\mathrm{E}[N_j] = \sum_{i=1}^{\infty} i \cdot \Pr[N_j = i] = \sum_{i=1}^{\infty} i(1-\nu)^{i-1}\nu = \frac{1}{\nu},$$

so,

$$\begin{aligned}\mathrm{E}[N|J=t] &\leq \frac{1}{\nu} \sum_{j=1}^{t} \big(1 + 20\alpha(y)^j \sigma(x,y)\big) \\ &< 21 \cdot \frac{\sigma(x,y)}{\nu} \sum_{j=1}^{t} \alpha(y)^j < 21 \cdot \frac{\sigma(x,y)}{\nu} \cdot \frac{\alpha(y)^{t+1}}{\alpha(y)-1}. \end{aligned} \tag{9}$$

Now, (1), (2), (8) and (9) give us

$$\begin{aligned}\mathrm{E}[N] &= \sum_{t=0}^{\infty} \mathrm{E}[N|J=t] \cdot \Pr[J=t] \\ &\leq \sum_{t<l} \mathrm{E}[N|J=t] + \sum_{t \geq l} \mathrm{E}[N|J=t] \cdot \Pr[J \geq t] \\ &< 21 \cdot \frac{\sigma(x,y)}{\nu} \cdot \left(\sum_{t=0}^{l-1} \frac{\alpha(y)^{t+1}}{\alpha(y)-1} + \sum_{t \geq l} \Big(\frac{\alpha(y)^{t+1}}{\alpha(y)-1} \cdot \gamma(y)^{t-l} \Big) \right) \\ &< 21 \cdot \frac{\sigma(x,y)}{\nu} \cdot \frac{\alpha(y)^{l+1}}{\alpha(y)-1} \cdot \Big(\frac{1}{\alpha(y)-1} + \frac{1}{1-\alpha(y)\gamma(y)} \Big) \\ &\leq 21 \cdot \frac{x}{x-1} \cdot \sigma(x,y) \cdot \frac{\alpha(y)^2}{\alpha(y)-1} \cdot \Big(\frac{1}{\alpha(y)-1} + \frac{1}{1-\alpha(y)\gamma(y)} \Big) \cdot \frac{Q}{\epsilon} \\ &= 21 \cdot \frac{x}{x-1} \cdot \sigma(x,y) \cdot \frac{(1+\gamma(y))^3}{\gamma(y)(\gamma(y)-1)^2} \cdot \frac{Q}{\epsilon}.\end{aligned}$$

The result follows.